

UNIVERSIDAD DE CUENCA
Facultad de Ingeniería



**“Una Arquitectura de Integración Tecnológica de Internet de las
Cosas y Computación en la Nube”**

Carrera de Ingeniería de Sistemas

Trabajo de Titulación Previo a la Obtención del Título de Ingeniero
de Sistemas

AUTORES:

Christian David Piedra García - CI: 0105785794
Pedro Alexander Tenezaca Sari - CI: 0105952360

DIRECTOR:

Ing. Irene Priscila Cedillo Orellana, PhD. - CI: 0102815842

CUENCA-ECUADOR
2018



RESUMEN

La Computación en la Nube (Cloud Computing) constituye una solución atractiva debido a sus características de calidad propias e indiscutiblemente interesantes, que solventan de una manera efectiva las necesidades de disponibilidad, rendimiento, escalabilidad y elasticidad con una mínima inversión de mantenimiento e infraestructura por su modelo de facturación “pay as you go” o pago únicamente por los servicios consumidos. Tras la aparición del Internet de las Cosas (IoT) se hace esencial conocer las ventajas y desventajas cuando se combinan estas dos tecnologías con el fin de lograr soluciones adecuadas que reúnan todos los requisitos tanto funcionales como no funcionales.

En este sentido, este trabajo de titulación, busca plantear el diseño de una arquitectura tecnológica siguiendo los lineamientos del paradigma CloudIoT aplicándolo a un dispositivo de tecnología asistiva que permita ayudar a uno de los grupos más vulnerables como son los adultos mayores. Para ello se estudiará cuál es el estado del arte en este contexto y se propondrá una arquitectura basada en las necesidades encontradas.

Palabras clave: Computación en la Nube, Internet de las Cosas - IoT.



ABSTRACT

Cloud Computing is an attractive solution due to its interesting quality characteristics, which effectively solve the needs of availability, performance, scalability and elasticity, with a minimum investment of maintenance and infrastructure because of its billing model " pay as you go ". After the appearance of the Internet of Things (IoT), it becomes imperative to know the advantages and disadvantages when combining these two technologies in order to achieve quality solutions.

In this context, this work seeks to propose the design of a technological architecture following the guidelines of the CloudIoT paradigm, applying it to an assisted technology device that allows one of the most vulnerable groups to be assisted, such as the elderly. For this, the state of the art in this context will be studied and an architecture based on the needs found will be proposed.

Keywords: Cloud Computing, Internet of Things - IoT.



ÍNDICE DE CONTENIDOS

RESUMEN	2
ABSTRACT	3
ÍNDICE DE CONTENIDOS	4
ÍNDICE DE FIGURAS	8
ÍNDICE DE TABLAS	9
Agradecimientos	14
Dedicatoria	15
Dedicatoria	16
Capítulo 1 Introducción	17
1.1 Motivación y Contexto	17
1.2 Planteamiento del problema	19
1.3 Solución propuesta	20
1.3.1 Hipótesis y Objetivos	20
1.3.1.1 Hipótesis	20
1.3.1.2 Objetivos	20
1.3.1.2.1 Objetivo General	20
1.3.1.2.2 Objetivos Específicos	20
1.4 Contexto del proyecto	21
1.5 Metodología de la Investigación	21
1.6 Estructura del trabajo de titulación	23
Capítulo 2 Marco Teórico	25
2. 1 Computación en la Nube (<i>Cloud Computing</i>)	25



2.1.1 Actores	26
2.1.2 Características Esenciales	26
2.1.3 Modelo en capas de computación en la nube	27
2.1.4 Modelos de Servicio de Computación en la Nube	28
2.2 Internet de las Cosas (<i>IoT</i>)	29
2.2.1 Dispositivos	30
2.2.2 Actuador	30
2.2.3 Aplicaciones	30
2.2.4 Recursos	31
2.2.5 Servicios	31
2.2.6 Características fundamentales	31
2.2.7 Modelo de referencia de IoT	32
2.2.7.1 Dispositivos físicos y controladores	33
2.2.7.2 Conectividad	33
2.2.7.3 Edge (Fog) Computing	33
2.2.7.4 Almacenamiento de datos	34
2.2.7.5 Abstracción de Datos	34
2.2.7.6 Aplicación	34
2.2.7.7 Colaboración y procesos	35
2.2.8 Protocolos para comunicación	35
2.7.8.1 HTTP	35
2.7.8.2 REST	35
2.7.8.3 MQTT (Messaging Queuing Telemetry Transport)	36
2.3 Integración de Computación en la Nube e IoT (CloudIoT)	36
2.3.1 Aplicaciones	38
2.3.1.1 Cuidado de la salud	38
2.3.1.2 Ciudades inteligentes (Smart Cities)	39
2.3.1.2.1 <i>Hogar inteligente y medición inteligente</i>	39
2.3.1.2.2 <i>Video vigilancia</i>	39
2.3.1.3 <i>Movilidad inteligente</i>	40
2.3.1.4 <i>Energía inteligente y red inteligente</i>	40
2.4 Lenguajes de dominio específico (<i>Domain Specific Language - DSL</i>)	40
2.5 Ingeniería Dirigida por Modelos (<i>Model Driven Engineering - MDE</i>)	40
2.6 Modelo de Dominio Específico (Domain Specific Model- DSM)	41
Capítulo 3 Estado del Arte	43



3.1 Integración Computación en la Nube e IoT (CloudIoT)	43
3.1.1 Cuidado de la Salud (<i>Healthcare</i>)	50
3.2 Discusión	55
Capítulo 4 Modelo de Dominio Específico (Mod4CIoT) y Lenguaje de Dominio Específico para Arquitecturas CloudIoT	56
4.1 Componentes Mod4CIoT	59
Capítulo 5 Arquitectura de integración de Computación en la Nube e IoT. Caso de Estudio: Ambient Assisted Living.	63
5.1 Caso de estudio: <i>Ambient Assisted Living (AAL)</i>	66
5.1.1 Descripción y contextualización del caso de estudio	66
5.2 Requisitos del sistema	68
5.3 Proceso de diseño de la arquitectura	68
5.3.1 Revisar las entradas	69
5.3.2 Establecer el objetivo de iteración	71
5.3.3 Elegir uno o más elementos del sistema para refinar	71
5.3.4 Elegir uno o más conceptos de diseño que satisfagan los factores seleccionados	71
5.3.5 Creación de instancias de elementos arquitectónicos, asignación de responsabilidades y definición de interfaces	72
5.3.6 Vistas de boceto y registro de las decisiones de diseño	73
Capítulo 6 Evaluación de la arquitectura propuesta	77
6.1 Introducción	77
6.2 Modelos teóricos de evaluación en Ingeniería de Software	78
6.2.1 Technology acceptance model (TAM)	78
6.2.2 Method Evaluation Model (MEM)	79
6.3 Adaptando el MEM para evaluar Arquitecturas CloudIoT	80
6.4 Evaluando la utilidad percibida de la Arquitectura CloudIoT la práctica: cuasi-experimento.	84
6.4.1 Definición de objetivos y preguntas de investigación	85
6.4.2 Diseño y Planificación del cuasi-experimento	87
6.4.2.1. Selección del contexto	87
6.4.2.2. Tareas experimentales	87
6.4.2.3 Variables	88
6.4.2.4 Material experimental	89
6.4.3 Colección de la evidencia del cuasi-experimento	90
6.4.4 Análisis de los datos del experimento	90
6.4.4.1 Análisis de las Percepciones de Usuario	90



6.4.4.2 Análisis del Rendimiento del Usuario	92
6.4.4.3 Análisis de las Relaciones Causales	92
6.4.4.4 Eficiencia vs Facilidad de Uso Percibida	93
6.4.4.5 Efectividad vs Utilidad Percibida	93
6.4.4.6 PEOU vs Utilidad Percibida	94
6.4.4.7 Intención de Uso vs Utilidad Percibida	95
6.4.4.8 Intención de Uso vs. Facilidad de Uso Percibida	95
6.5 Amenazas a la validez	96
6.5.1 Validez interna	96
6.5.2 Validez externa	96
6.5.3 Validez del constructo	97
6.5.3 Validez de la conclusión	97
6.7 Conclusiones	97
Capítulo 7 Conclusiones y trabajos futuros	98
7.1 Conclusiones	98
7.1.1 Objetivo general	98
7.1.2 Objetivos específicos	99
7.1.2.1 Objetivo específico 1	99
7.1.2.2 Objetivo específico 2	99
7.1.2.3 Objetivo específico 3	100
7.1.2.3 Objetivo específico 4	100
7.1.3 Conclusiones generales	100
7.2 Trabajo Futuro	101
REFERENCIAS BIBLIOGRÁFICAS	102
Anexos	109
Anexo A: Consentimiento informado	109
Anexo B: Ejercicio	111
Anexo 1. Modelo y Tipos de Datos	114
Arquitectura Original	118
Arquitectura Modificada	119

ÍNDICE DE FIGURAS

Figura 1-1: Modelo de investigación cuantitativo Fuente: (Hernández, Fernández y Baptista, 2014).....	21
Figura 1-2: Estructura del trabajo de titulación. Fuente: Elaboración propia.	24
Figura 2-1 Arquitectura de Computación en la Nube. Fuente: Jing y Jian-jun (2010).....	25
Figura 2-2 Características y alcance de un sistema IoT. Fuente: IEEE-Internet of Things (Minerva et al., 2015).....	30
Figura 2-3 Modelo de Referencia IOT CISCO. Fuente: (Green, 2014).	32
Figura 2-4 Arquitectura MDE. Fuente: Arquitectura Ingeniería Dirigida por Modelos (Kurtev, 2005).....	41
Figura 3-1 Pasarela Inteligente con Fog Computing. Fuente: Aazam y Huh (2014).....	43
Figura 3-2 Arquitectura en Capas de Pasarela Inteligente con Fog Computing. Fuente: (Aazam y Huh, 2014).	44
Figura 3-3 Esquema arquitectónico y módulos. Fuente: Distefano et al. (2013).....	45
Figura 3-4 Arquitectura GaaS. Fuente: Wu et al. (2012)	45
Figura 3-5 Pasarela Restful Web. Fuente: Wu et al. (2012)	46
Figura 3-6 Marco general propuesto para la nube de IoT en el escenario de cuidado de la salud. Fuente: Hassan et al. (2014)	47
Figura 3-7 Arquitectura proyecto SIGMA. Fuente: Puliafito (2014).....	48
Figura 3-8 Escenario de Integración. Fuente: Puliafito (2014)	48
Figura 3-9 Arquitectura CloudThings: Plataforma IoT basada en la nube. Fuente: (Zhou et al., 2013).....	49
Figura 3-10 Arquitectura de Computación en la Nube e IoT orientada a dispositivos médicos. Fuente: Doukas y Maglogiannis (2012)	51
Figura 3-11 Arquitectura propuesta para gestionar los datos de atención médica generalizada en la nube. Fuente: Doukas et al. (2012)	52
Figura 3-12 Arquitectura en capas propuesta. Fuente: Mohammed et al. (2014)	53
Figura 4-1 DSM Mod4CIoT y enumeraciones correspondientes. Fuente: Elaboración propia...	58
Figura 5-1 Arquitectura general CloudIoT. Fuente: Elaboración propia.....	65
Figura 5-2 Proceso de diseño de la una arquitectura CloudIoT. Fuente: Elaboración propia.....	69
Figura 5-3 Arquitectura obtenida del caso de estudio. Fuente: Elaboración propia.....	74
Figura 6-1 Technology Acceptance Model (TAM) simplificado. Fuente: Riemenschneider, Hardgrave, & Davis (2002).....	78
Figura 6-2 Method Evaluation Model – MEM. Fuente: Moody (2001)	79
Figura 6-3 Distribución de preguntas del cuestionario aplicado al cuasi-experimento. Fuente: Elaboración propia.	81
Figura 6-4 Modelo teórico para la evaluación de arquitecturas CloudIoT. Fuente: Elaboración propia.	82
Figura 6-5 Descripción general del proceso de investigación que incluye entradas y salidas para cada fase. Fuente: Elaboración propia.....	85



Figura 6-6 Diagrama de cajas para las variables PEOU, PU e ITU. Cuasi-Experimento. Fuente: Elaboración propia. 91

ÍNDICE DE TABLAS

Tabla 2-1 Características Esenciales de la Computación en la Nube. Fuente: (Mell et al., 2011)	27
Tabla 2-2 . Modelos de Despliegue de Computación en la Nube. Fuente: (Mell et al., 2011).....	29
Tabla 2-3 Características de IoT. Fuente: IEEE-Internet of Things (Minerva et al., 2015).....	32
Tabla 2-4 Características complementarias Computación en la Nube e IoT. Fuente: (Botta et al., 2014).....	37
Tabla 2-5 Factores para la integración CloudIoT. Fuente: (Botta et al., 2016).	38
Tabla 3-1 Tabla comparativa revisión estado del arte. Fuente: Elaboración propia.	54
Tabla 4-1 Componentes del DSM. Fuente: Elaboración propia.	62
Tabla 5-1 Entradas del proceso de diseño. Fuente: Elaboración propia.	71
Tabla 5-2 Conceptos de diseño seleccionados. Fuente: Elaboración propia.	72
Tabla 5-3 Casos de uso del sistema. Fuente: Elaboración propia.	73
Tabla 5-4 Casos de uso del sistema. Fuente: Elaboración propia.	75
Tabla 6-1 Ítems definidos para medir las variables basadas en la percepción. Fuente: Elaboración propia.	84
Tabla 6-2 Meta del experimento de acuerdo al paradigma Goal-Question Metric (GQM). Fuente: Elaboración propia.	85
Tabla 6-3 Variables dependientes basadas en la percepción. Fuente: Elaboración propia.	88
Tabla 6-4 Variables dependientes basadas en el rendimiento. Fuente: Elaboración propia.....	89
Tabla 6-5 Prueba de Shapiro-Wilk para las variables subjetivas. Fuente: Elaboración propia..	91
Tabla 6-6 Prueba t con una muestra para las variables subjetivas. Fuente: Elaboración propia.	92
Tabla 6-7 Estadística Descriptiva para Variables Basadas en la percepción del Usuario.	92
Tabla 6-8 Niveles de significancia. Fuente: (Moody, 2001)	92
Tabla 6-9 Regresión Simple entre la Eficiencia Actual y la Facilidad de Uso Percibido. Fuente: Elaboración propia.	93
Tabla 6-10 Regresión simple entre la Efectividad Actual y la Utilidad Percibida. Fuente: Elaboración propia.	94
Tabla 6-11 Regresión Simple entre la Facilidad de Uso Percibida y la Utilidad Percibida. Fuente: Elaboración propia.	94
Tabla 6-12 Regresión Simple entre Utilidad Percibida e Intención de Uso. Fuente: Elaboración propia.	95
Tabla 6-13 Regresión Simple entre Facilidad de Uso Percibida e Intención de Uso. Fuente: Elaboración propia.	96



Cláusula de licencia y autorización para publicación en el Repositorio Institucional

Christian David Piedra García, en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación “Una Arquitectura de Integración Tecnológica de Internet de las Cosas y Computación en la Nube”, de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 26 de octubre de 2018

Christian David Piedra García

C.I. 0105785794



Cláusula de Propiedad Intelectual

Christian David Piedra García, autor del trabajo de titulación “Una Arquitectura de Integración Tecnológica de Internet de las Cosas y Computación en la Nube”, certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor/a.

Cuenca, 26 de octubre de 2018

Christian David Piedra Garcia

C.I: 0105785794



Cláusula de licencia y autorización para publicación en el Repositorio Institucional

Pedro Alexander Tenezaca Sari, en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación “Una Arquitectura de Integración Tecnológica de Internet de las Cosas y Computación en la Nube”, de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 26 de octubre de 2018

Pedro Alexander Tenezaca Sari

C.I. 0105952360



Cláusula de Propiedad Intelectual

Pedro Alexander Tenezaca Sari, autor del trabajo de titulación "Una Arquitectura de Integración Tecnológica de Internet de las Cosas y Computación en la Nube", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor/a.

Cuenca, 26 de octubre de 2018

Pedro Alexander Tenezaca Sari

C.I: 0105952360



Agradecimientos

A nuestra directora, Ing. Priscila Cedillo por su apoyo, guía y consejos a lo largo de este trabajo de titulación.

A la Comunidad Estudiantil de Microsoft y a la Rama Estudiantil IEEE por ayudarnos a crecer personal y profesionalmente a lo largo de nuestra vida universitaria.

A nuestros compañeros de carrera por ayudarnos con la evaluación de esta tesis.

A nuestros amigos y familiares, por ser parte de nuestra vida y alentarnos a continuar cada día y convertir el trabajo diario en alegría.

A la Facultad de Ingeniería, a la Universidad de Cuenca por prepararnos como buenos profesionales.



Dedicatoria

A mi padre Nardo†, que en paz descanse, quien fue mi ejemplo a seguir y me enseñó que una persona no renuncia a sus sueños y metas por más difíciles que parezcan.

A mi madre Catalina, que desde la distancia, siempre me ha apoyado en mis decisiones y a seguir siempre adelante.

A mi abuelita Cecilia, a mi tía Marcela, quienes han tenido el papel de Padre y Madre a lo largo de mi vida, me han apoyado siempre y me han impulsado cada día a ser una mejor persona, estaré eternamente agradecido.

A mi hermano Mateo, a mi hermana Belén, a mi sobrino Gabriel, a mi primo Emilio, son mi apoyo y motivación.

A mi compañero de tesis y amigo Pedro, a Steven, a Alex, a Samuel, a Valeria, con quienes he descubierto la amistad, la lealtad y con quienes he pasado malos y también los mejores momentos de mi vida universitaria, porque al trabajar en equipo se llega más lejos que estando solo.

A Priscila Cedillo, directora de esta tesis, por apoyarnos siempre y querernos como si fuéramos sus hijos y amigos. Por alentarnos a seguir adelante y a siempre pensar en grande, por confiar en nosotros, guiarnos en este trabajo y enseñarnos a crecer a nivel personal y profesional.

A mis amigas y amigos de la Rama Estudiantil IEEE, quienes han sido una parte importante de mi vida, siempre los llevaré en mis pensamientos.

A la Universidad de Cuenca, por ser mi segundo hogar. A mis compañeros y docentes, porque gracias a sus enseñanzas he logrado culminar mi carrera.

David



Dedicatoria

A mis padres Cecilia y Angel, quienes me han guiado y apoyado constantemente para cumplir con los objetivos que me he propuesto.

A mis hermanas María Paz, Juliana, a mi sobrina Paula, a mis hermanos Gonzalo y Xavier, son mi motivación.

A mi compañero de tesis David, a Vale, Steven y Alex junto a quien hemos compartido malos, buenos y excelentes momentos durante nuestra carrera universitaria, porque el trabajo en equipo siempre trae los mejores resultados.

A nuestra directora Priscila, por siempre confiar en nosotros. Por alentarnos en el proceso de desarrollo del presente trabajo y nuestro ejemplo para mejorar a nivel personal y profesional.

A mis amigos y amigas de la Comunidad Estudiantil de Microsoft, de la Rama Estudiantil IEEE, por permitirme ser parte de su vida.

A la Universidad de Cuenca, a mis compañeros de carrera y docentes, porque con sus enseñanzas nosotros los estudiantes mejoramos cada día.

Pedro



Capítulo 1

Introducción

1.1 Motivación y Contexto

El modelo de Computación en la Nube (Cloud Computing) se refiere al alojamiento y la prestación de servicios a través de Internet el cual posee virtualmente capacidades ilimitadas en términos de almacenamiento y procesamiento. Esta tecnología es atractiva para los propietarios de negocios, pues permite a las empresas consumir los recursos bajo demanda de sus servicios (Zhang, Cheng y Boutaba, 2010).

El paradigma de Internet de las Cosas (IoT) tiene como idea básica la presencia ubicua a nuestro alrededor de una variedad de cosas u objetos, (p. ej. etiquetas de identificación de radiofrecuencia - RFID, sensores, actuadores, teléfonos móviles) capaces de interactuar entre ellos y cooperar con sus vecinos para alcanzar objetivos comunes (Atzori, Iera y Morabito, 2010; Eleftherakis, Pappas, Lagkas, Rousis y Paunovski, 2015). Generalmente los dispositivos IoT se caracterizan por tener limitaciones en el almacenamiento de información y la capacidad de procesamiento; como consecuencia de esto, existen problemas en la confiabilidad, rendimiento, seguridad y privacidad (Botta, De Donato, Persico, y Pescapé, 2014). Además, IoT tiene aplicaciones en escenarios empresariales, en el hogar (p. ej., ambientes de vida asistidos, domótica, *e-health*, transporte inteligente, ciudades inteligentes) e influye directamente sobre la industria (p. ej. logística, automatización industrial, seguridad) (Botta, Donato, Persico y Pescapé, 2016).

Por otra parte, dentro del contexto de la elaboración de este trabajo de fin de carrera, se ha contemplado el concepto de ambientes de vida asistidos (Ambient Assisted Living - AAL); éste se está convirtiendo en un campo de investigación activo subsumido por las tecnologías de Inteligencia Ambiental (Ambient Intelligence - Aml) debido al cambio demográfico en los países industrializados en todo el mundo, lo que causa un gran incremento de personas mayores y con capacidades especiales que demandan más servicios de asistencia médica; solicitando al mismo tiempo acceso rápido y confiable a los sistemas sociales, médicos y de emergencia (Steg, Strese, Hull y Schmidt, 2005).



Las tecnologías asistivas se han convertido en un campo importante en ciencias médicas. La Organización Mundial de la Salud (OMS) define la Tecnología de Asistencia (AAT) "Como sistemas y servicios relacionados con la entrega de productos de apoyo que permiten a las personas vivir vidas saludables, productivas, independientes y dignas, y también pueden participar en la educación, Mercado laboral y vida cívica " (Health Organization World, 2017). Los grupos prioritarios en el área médica (pueden variar según la ubicación); éstos son: mujeres embarazadas, individuos con discapacidades intelectuales y de desarrollo, necesidades especiales, personas con enfermedades catastróficas, niños y adultos mayores (Ministerio de Salud Pública del Ecuador, 2012). Todos ellos podrían beneficiarse de la tecnología asistiva, para reducir la necesidad de servicios de salud formales. Entonces, para el año 2050, más de 2.000 millones de personas necesitarán al menos un producto que les asista, ayudando a personas pertenecientes a grupos prioritarios, especialmente a adultos mayores (Garçon et al., 2016).

La falta de disponibilidad de información, relacionada con los pacientes, causa muchos errores en la asistencia de la salud. El uso de las nuevas tecnologías de la información y la comunicación (TIC) podría aumentar la accesibilidad de la información médica y es esencial para la seguridad del paciente (Turcu y Turcu, 2013). Por otro lado, el IoT es una infraestructura de red global, que enlaza objetos físicos y virtuales a través de la explotación de capacidades de captura de datos y comunicaciones (ITU, 2005). La conectividad de los sensores y otros dispositivos para asistencia médica juegan un papel importante en el cuidado de los pacientes, ya que permiten acceder en tiempo real a la información médica.

Además, con el desarrollo de la web y con el fin de aumentar las capacidades de aplicaciones en IoT, se ha intentado integrar junto a Cloud Computing, definiendo un nuevo paradigma llamado CloudIoT (Botta et al., 2016). En general, IoT puede beneficiarse de las capacidades y recursos prácticamente ilimitados de Cloud para compensar sus limitaciones tecnológicas (por ejemplo, almacenamiento, procesamiento, comunicación); además, Cloud Computing ofrece una solución eficaz para la gestión y composición de servicios de IoT, así como para implementar aplicaciones y servicios que explotan las cosas o los datos producidos por ellos (Lee et al., 2010).

Una arquitectura Cloud, al igual que una arquitectura IoT están estructuradas en capas (Tan, 2010; Jadeja, 2012) y corresponden a aplicaciones o situaciones específicas (p. ej. medicina, agricultura, estaciones meteorológicas, entre otras). Además, en algunos casos, una arquitectura puede expresarse y definirse mediante el uso de un Lenguaje de Dominio Específico (Domain

Specific Language - DSL), el cual consiste un lenguaje de programación con expresiones limitadas, enfocado en un dominio en particular (Fowler, 2010).

Algunas arquitecturas proponen soluciones mediante Fog Computing (Aazam & Huh, 2014) o Gateway como Servicio (Wu et al., 2012), pero no consideran la integración de sistemas externos. Distefano, Merlino y Puliafito (2013), presentan una arquitectura basada en infraestructura como servicio; sin embargo cada componente en la arquitectura es estrictamente necesario, lo que no la hace dinámica. En el trabajo (Hassan, Albakr, & Al-dossari, 2014), en el estudio presentado por Puliafito (2014) y en el trabajo presentado por (Zhou et al., 2013) se propone un marco de IoT asistido por la nube, para abordar los problemas desafiantes en el dominio de cuidado de salud e integración de sistemas externos, pero no se define un Lenguaje de Dominio Específico que permita un fácil entendimiento para los expertos en el dominio de AAL y por desarrolladores. Adicionalmente, arquitecturas relacionadas a cuidado de la salud presentadas por Doukas y Maglogiannis (2012), y Doukas, Pliakas, Tsanakas y Maglogiannis (2012) agregan el uso de dispositivos médicos y *wereables*, sin embargo no analizan la integración con servicios y sistemas externos como hospitales o centros de cuidado. Finalmente, se puede decir que si bien existen soluciones que permiten definir una arquitectura de Internet de las Cosas o de Cloud Computing (Lee et al., 2010), en dominios específicos (p. ej. AAL, ciudades inteligentes, healthcare) y a su vez se definen DSLs, no existe una forma estándar de generar una arquitectura de integración tecnológica de Internet de las Cosas y Computación en la Nube que cubra las características de las arquitecturas antes mencionadas.

1.2 Planteamiento del problema

La falta de disponibilidad de información que se relaciona con los pacientes causa errores en la asistencia de la salud, y aunque existen estudios en las tecnologías de Cloud Computing e IoT (CloudIoT) y cómo integrarlas, no existe una arquitectura estandarizada que permita a usuarios y empresas integrarlos de manera óptima en al área de tecnología asistida, es por ello que se necesitan arquitecturas que, desde el punto de vista de la ingeniería del software, permitan integrar componentes en aplicaciones CloudIoT. Además, las arquitecturas deberían integrar niveles de seguridad debería permitir ser evaluada según su prioridad.

Por ello, se debería contribuir con una investigación clara y fundamentada en normas estandarizadas, que permita determinar la estructura básica de una arquitectura que solvante los principales problemas y necesidades de CloudIoT. A pesar de que existen publicaciones que proponen soluciones para construir arquitecturas Cloud e IoT, no existen lenguajes de dominio específico que permitan definir correctamente una arquitectura CloudIoT.

1.3 Solución propuesta

En el presente trabajo se estudia el estado del arte y se plantea el diseño de una arquitectura tecnológica siguiendo los lineamientos del paradigma CloudIoT aplicándolo a un ambiente de tecnología asistiva.

Además, se ha definido un metamodelo que integra los diferentes componentes necesarios para generar una arquitectura CloudIoT y DSM con base en el metamodelo, el cual permite generar la arquitectura semánticamente, de modo que tanto desarrolladores y expertos en el dominio, pueden manejarla.

Finalmente, se ha generado un caso de estudio, con un escenario de la vida real con los principales actores y componentes que deberían interactuar para cumplir con los requisitos del caso en el dominio de AAL, sobre la cual se ha realizado una evaluación de la arquitectura generada para demostrar empíricamente que esta arquitectura ha realizado un aporte en investigación

1.3.1 Hipótesis y Objetivos

A continuación, se definen las hipótesis en base a las preguntas de investigación y los objetivos.

1.3.1.1 Hipótesis

- El uso de DSM facilita la construcción de arquitecturas CloudIoT en dominios específicos.
- Es posible diseñar una arquitectura que integre Computación en la Nube e Internet de las Cosas como solución a un escenario de AAL y sus requisitos.

1.3.1.2 Objetivos

Para el desarrollo de este trabajo de titulación se ha definido un objetivo general y varios objetivos específicos.

1.3.1.2.1 Objetivo General

Investigar y analizar los componentes que deben ser considerados al momento de combinar la Computación en la Nube (Cloud Computing) con el Internet de las Cosas (IoT), a fin de proponer el diseño de una arquitectura tecnológica siguiendo los lineamientos del paradigma CloudIoT aplicándolo a un ambiente de tecnología asistido.

1.3.1.2.2 Objetivos Específicos

- Estudiar el estado del arte relacionado con la integración de las tecnologías

de Computación en la Nube e IoT.

- Definir un lenguaje y un modelo de dominio específico (DSL, DSM) que permita representar una arquitectura CloudIoT.
- Proponer una arquitectura que integre las tecnologías de Computación en la Nube e IoT en campo de tecnología asistiva y aplicarlo a un escenario real.
- Evaluar la solución en un dominio específico con la ayuda de profesionales en formación.

1.4 Contexto del proyecto

Este trabajo de titulación se ha desarrollado en el contexto del grupo de investigación de “Evaluación del conocimiento sobre la alimentación, ejercicio, redes sociales y desarrollo de prototipos para la toma de medicación y monitorización de signos vitales orientados al adulto mayor mediante la integración y el uso de Tecnologías de la Información y Comunicación, aplicado a personas mayores de 65 años”, proyecto de CEPRA X-2016-04, dirigido por la Dra. Priscila Cedillo. Este proyecto integra varias instituciones universitarias; Universidad de Cuenca (Facultad de Ciencias Médicas y Facultad de Ingeniería), Escuela Politécnica del Ejército (ESPE), Universidad Católica de Cuenca y Universidad del Azuay.

1.5 Metodología de la Investigación

La metodología de investigación que se siguió, es de tipo cuantitativo y se compone de 10 fases. En la Figura 1.1 se describe el modelo de investigación seguida (Hernández, Fernández y Baptista, 2014).

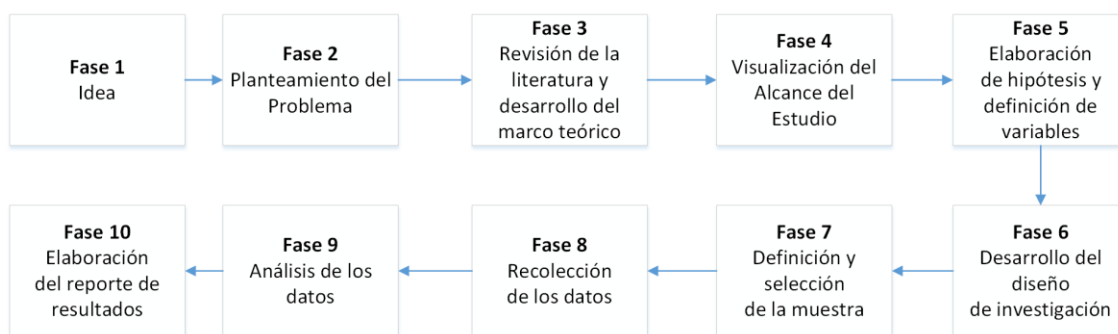


Figura 1-1: Modelo de investigación cuantitativo Fuente: (Hernández, Fernández y Baptista, 2014)

Cada fase de la metodología ha sido dirigida al dominio estudiado, se definen de la siguiente manera:

1. **Idea:** En esta fase se han originado las ideas de investigación planteadas en el dominio de Computación en la Nube e Internet de las Cosas.

2. **Planteamiento del problema:** En esta fase planteamos una problemática que da origen al proyecto de investigación. El planteamiento de la problemática, que abarca: los objetivos que persigue la investigación, las preguntas de investigación y la justificación, se describe en el presente capítulo.
3. **Desarrollo de la base tecnológica y revisión del estado del arte:** Conociendo ya la problemática y su contexto, se ha procedido a levantar la sustentación teórica con el fin de presentar un trabajo autocontenido y posteriormente conocer los estudios que se están llevando a cabo en el campo del problema planteado su realidad y contexto. Por lo tanto, fue importante realizar una revisión de la literatura; para ello se realizó una revisión del estado del arte en arquitecturas CloudIoT.
4. **Visualización del alcance del estudio:** En esta fase decidió cuál es el alcance del estudio en base al estudio del estado del arte.
5. **Elaboración de hipótesis y definición de variables:** De acuerdo al alcance del proyecto hemos establecido las hipótesis que son guías de investigación. En base a las hipótesis definimos las variables a estudiar, que han sido observadas y medidas con el fin de poner a prueba las hipótesis.
6. **Desarrollo del diseño de la investigación:** Esta etapa refiere al plan que el investigador hace con el fin de obtener información destinada a verificar la certeza de las hipótesis planteadas. En esta fase proponemos un modelo de lenguaje específico, a partir del cual diseñamos una arquitectura en área de AAL.
7. **Definición y selección de la muestra:** En esta etapa se van a evaluar la arquitectura CloudIoT en el dominio de cuidado de la salud en base al escenario propuesto.
8. **Recolección de los datos:** En esta fase se ha realizado evaluaciones a estudiantes de la carrera de Ciencias de la Computación de la Universidad de Cuenca para evaluar características de la arquitectura.
9. **Análisis de los datos:** En esta fase se han analizado los datos recolectados mediante varias fases de análisis que comprende: seleccionar un software apropiado para analizar los datos, ejecutar dicho software, explorar los datos, evaluar la confiabilidad y validez logradas por los instrumentos de medición, analizar mediante pruebas estadísticas las hipótesis planteadas, se realiza análisis adicionales y finalmente se preparan los resultados para ser presentados.
10. **Elaboración del reporte de resultados:** En la fase final del método de investigación, se analiza el contexto y se presenta los resultados y documentación de cada fase, teniendo como consecuencia el presente trabajo de titulación.

1.6 Estructura del trabajo de titulación

La Figura 1-2 muestra la estructura del trabajo de titulación, relacionando cada capítulo con cada fase de la metodología de investigación de Hernández, Fernández y Baptista (2014): Se describe cada de manera sucinta cada capítulo a continuación:

- **Capítulo 1. Introducción:** Se realiza la presentación del trabajo dando una idea amplia sobre la motivación para realización de este estudio, objetivo general, objetivos específicos, problemática abordada y los contenidos a desarrollar.
- **Capítulo 2. Marco teórico:** Se muestra todos los conceptos necesarios para un mejor entendimiento del lector de los temas a tratar, como es el significado de Computación en la Nube, Internet de las Cosas, paradigma CloudIoT, *Domain-specific modeling*, entre otros.
- **Capítulo 3. Estado del arte:** Incluye el estado actual de la investigación que abarca las arquitecturas resultantes de la combinación de las dos tecnologías (Computación en la Nube, Internet de las Cosas).
- **Capítulo 4. Modelo de Dominio Específico (Mod4CloT) y Lenguaje de Dominio Específico para Arquitecturas CloudIoT:** Se presenta el modelo de dominio específico Mod4CloT para el diseño arquitecturas de sistemas CloudIoT, así como, la motivación para su creación.
- **Capítulo 5. Arquitectura de integración de Computación en la Nube e IoT. Caso de Estudio Ambient Assisted-Living:** Se propone una arquitectura ClouIoT en base a un caso de estudio de ambiente de vida asistido. Para este propósito se ha seguido un método para el diseño de arquitecturas y se ha utilizado Mod4CloT.
- **Capítulo 6. Evaluación de la arquitectura propuesta:** Se presenta la evaluación de la arquitectura propuesta mediante un cuasi-experimento a estudiantes de la carrera de Ciencias de la Computación de la Universidad de Cuenca.
- **Capítulo 7. Conclusiones y trabajo futuro:** Se presenta las consecuencias obtenidas al finalizar el presente trabajo de titulación, así como el trabajo futuro a realizar.
- **Anexos:** Contiene cada uno de documentos que han sido generados para realizar las evaluaciones de la arquitectura propuesta.

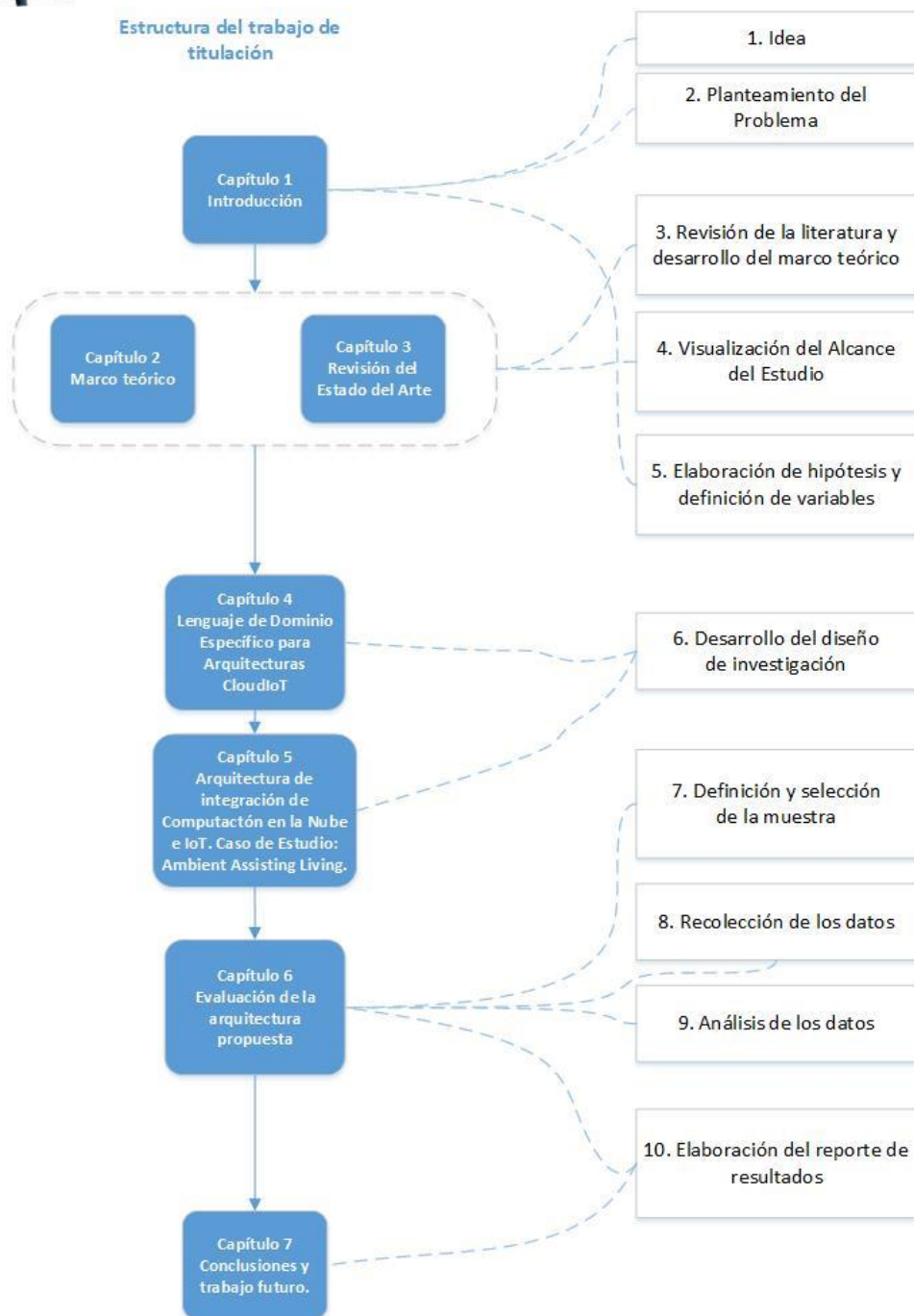


Figura 1-2: Estructura del trabajo de titulación. Fuente: Elaboración propia.

Capítulo 2

Marco Teórico

El marco teórico se fundamenta en varios pilares que son las bases para el entendimiento del presente trabajo de titulación, estos son: la conceptualización de Computación en la nube, Internet de las cosas y paradigma CloudIoT.

2.1 Computación en la Nube (*Cloud Computing*)

La Computación en la Nube (Cloud Computing) es un modelo que permite, el acceso ubicuo a la red bajo demanda a un grupo compartido de recursos informáticos configurables (por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios) que pueden aprovisionarse y liberarse rápidamente con un mínimo esfuerzo administrativo o interacción del proveedor de servicios (Mell, Grance y Grance, 2011), concepto definido según el Instituto Nacional de Estándares y Tecnología (NIST). La Computación en la Nube no es una tecnología nueva, sino un nuevo modelo de operaciones que reúne un conjunto de tecnologías existentes (e.g virtualización, fijación de precios basada en servicios) accesibles a través de Internet, los usuarios pueden acceder a los servicios de la nube pagando solo por el tiempo y los servicios que necesitan (Zhou et al., 2013). La Figura 2-1 muestra la arquitectura de un ambiente de Computación en la Nube dividida en cuatro capas.

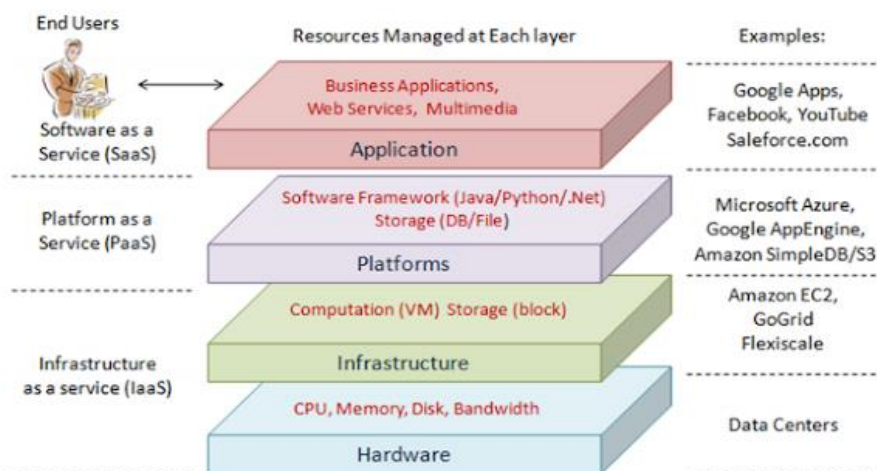


Figura 2-1 Arquitectura de Computación en la Nube. Fuente: Jing y Jian-jun (2010).

2.1.1 Actores

Con base en el análisis de los proveedores de servicios de computación en la nube y basados en las capas del modelo de servicios de computación en la nube, en (Leimeister, Böhm, München, Riedl y München, 2010) identifican a los siguientes actores:

- *Cliente:* El cliente compra servicios a través de varios canales de distribución, por ejemplo, directamente del proveedor del servicio o a través de un proveedor de la plataforma.
- *Proveedor de servicios:* Los proveedores de servicios, también etiquetados como proveedores de TI, desarrollan y operan servicios que ofrecen valor al cliente y al proveedor de servicios agregado, respectivamente. Desarrollan aplicaciones que se ofrecen e implementan en la plataforma de computación en la nube y acceden al hardware y la infraestructura de los proveedores de infraestructura.
- *Proveedor de infraestructura:* Los proveedores de infraestructura proporcionan a la red de valor todos los servicios de computación y almacenamiento necesarios para ejecutar aplicaciones dentro de la nube y proporcionar la columna vertebral técnica. Ofrecen el hardware necesario y escalable para los servicios sobre el cual los proveedores de servicios ofrecen sus servicios.
- *Proveedor de servicios agregados:* Los proveedores de servicios agregados pueden considerarse como una forma especializada del proveedor de servicios, que ofrece nuevos servicios o soluciones combinando servicios preexistentes o partes de servicios para formar nuevos servicios y ofrecerlos a los clientes.
- *Proveedor de plataforma:* El proveedor de la plataforma ofrece un entorno dentro del cual se pueden implementar aplicaciones en la nube. Actúa como un tipo de catálogo en el que diferentes proveedores de servicios ofrecen servicios.
- *Consultante:* la consultoría para los clientes sirve como soporte para la selección e implementación de servicios relevantes para crear valor para su modelo de negocio.

2.1.2 Características Esenciales

Según el Instituto Nacional de Estándares y Tecnología (NIST) (Mell et al., 2011), el modelo de Computación en la Nube está compuesto por 5 características esenciales que se detallan a continuación:

Característica	Descripción
Servicio Bajo Demanda	Son las capacidades de computación que no requieren interacción humana con el proveedor de servicios para ser provisionados, como el tiempo del servidor y el almacenamiento en red.
Acceso Amplio a la Red	Los recursos están disponibles mediante el uso de la red y se accede a través de mecanismos que soportan la utilización de plataformas heterogéneas (p. ej., tablets, teléfonos móviles, computadoras portátiles y estaciones de trabajo).
Pila de Recursos	Los recursos informáticos del proveedor se agrupan para brindar un servicio a múltiples consumidores, mediante un modelo multiusuario con recursos físicos y virtuales que son asignados de acuerdo a la demanda del consumidor. Permite mantener múltiples clientes de los mismos recursos físicos, separando de forma segura los recursos en el nivel lógico.
Elasticidad Rápida	Los servicios en la nube se pueden aprovisionar de manera rápida para adicionar o disminuir recursos, bajo demanda o en algunos casos automáticamente. Para el consumidor, las capacidades disponibles para el aprovisionamiento parecen ser ilimitadas y accesibles en cualquier momento.
Servicio a la Medida	El uso de recursos puede ser monitoreado, controlado, mejorado, reportado automáticamente, valiéndose en su utilización. Proporcionando transparencia tanto para el proveedor como para el consumidor del servicio utilizado.

Tabla 2-1 Características Esenciales de la Computación en la Nube. Fuente: Mell et al. (2011)

2.1.3 Modelo en capas de computación en la nube

La arquitectura de la nube se puede dividir en cuatro capas: (i) centro de datos (hardware), (ii) infraestructura, (iii) plataforma y (iv) aplicación. Cada capa se puede ver como un servicio para la capa anterior y como un consumidor para la capa siguiente (Zhang et al., 2010).

2.1.1.1 La capa de hardware:

Esta capa es responsable de gestionar los recursos físicos de la nube, incluidos servidores físicos, enrutadores, conmutadores, sistemas de alimentación y refrigeración. En la práctica, la capa de hardware generalmente se implementa en centros de datos.

2.1.1.2 La capa de infraestructura:

También conocida como capa de virtualización. La capa de infraestructura es un componente esencial de la computación en la nube, permite la asignación



dinámica de recursos, solo disponibles a través de las tecnologías de virtualización.

2.1.1.3 La capa de plataforma:

Construido sobre la capa de infraestructura, la capa de la plataforma consiste en sistemas operativos y marcos de aplicaciones.

2.1.1.4 La capa de aplicación:

En el nivel más alto de la jerarquía, la capa de aplicación consiste en las aplicaciones de nube reales. A diferencia de las aplicaciones tradicionales, las aplicaciones en la nube pueden aprovechar la función de escalado automático para lograr un mejor rendimiento, disponibilidad y un menor costo operativo.

2.1.4 Modelos de Servicio de Computación en la Nube

La computación en la nube se puede clasificar en términos de los tipos de servicio y el rango de servicio. La Tabla 2-2 ilustra los diferentes modelos basados en las dos clasificaciones. (Mell et al., 2011)

Modelo de Computación en la nube		
Modelos de Servicio (Nivel/tipo de servicio)	Infraestructura como Servicio (IaaS)	Proporciona a los consumidores recursos informáticos: procesamiento, almacenamiento, redes y otros recursos fundamentales donde el consumidor puede implementar y ejecutar software arbitrario, que puede incluir sistemas operativos y aplicaciones.
	Plataforma como Servicio (PaaS)	Proporciona un entorno basado en la Nube con todos los requisitos necesarios para dar soporte a todo el ciclo de vida de creación y puesta en marcha de aplicaciones creadas por el consumidor, incluidos bibliotecas, servicios, soporte de sistema operativo y marcos de desarrollo de software.
	Software como Servicio (SaaS)	Aplicaciones bajo demanda ejecutándose sobre una infraestructura en la Nube o Cloud. Las aplicaciones son accesibles a través de Internet y, por lo general, de un navegador

web o una interfaz de programa.		
Modelos de despliegue (Rango/escala del servicio)	Nube Pública	Los proveedores de servicios ofrecen sus recursos como servicios al público en general. Puede ser propiedad, administrado y operado por una organización empresarial, académica o gubernamental.
	Nube Comunitaria	Infraestructura Cloud provisionada para uso exclusivo de una sola comunidad de consumidores de una organización que tienen objetivos en común.
	Nube Privada	Infraestructura Cloud provisionada para uso exclusivo de una sola organización, administrado y operado por la propia organización
	Nube Híbrida	Composición de dos o más infraestructuras de nube distintas. Parte de la infraestructura del servicio se ejecuta en una nube, mientras que la parte restante se ejecuta en otra.

Tabla 2-2 . Modelos de Despliegue de Computación en la Nube. Fuente: Mell et al. (2011)

2.2 Internet de las Cosas (IoT)

Según IEEE en su documento (Minerva, Abyi y Rotondi, 2015) que recopila estándares y proyectos en IoT, define a IoT como una red que conecta “cosas”, identificables de manera única en internet, las cuales tienen capacidades de detección, actuación y son potencialmente programables; por lo tanto al explotar estas características de identificación y detección únicas, la información sobre cada cosa puede ser recopilada y cambiada desde cualquier lugar, en cualquier momento y por cualquier “cosa”.

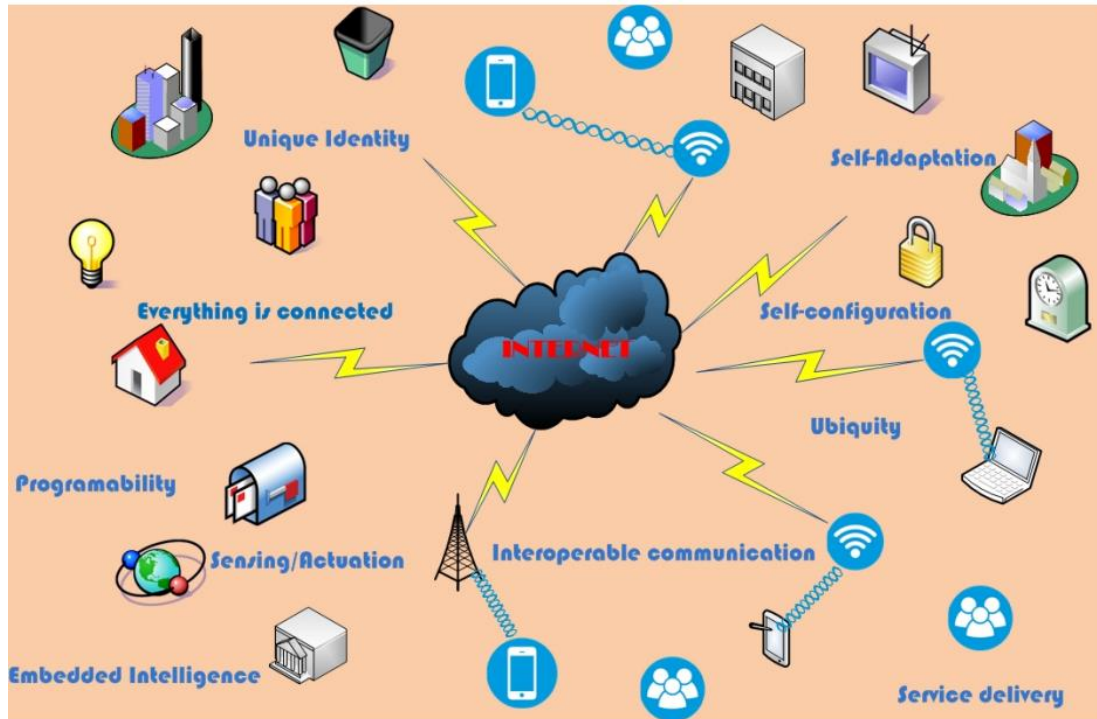


Figura 2-2 Características y alcance de un sistema IoT. Fuente: IEEE-Internet of Things (Minerva et al., 2015).

2.2.1 Dispositivos

Un dispositivo es un componente físico (hardware) con capacidades de comunicación que lo vinculan a otros sistemas de TI (Tecnologías de la Información). Un dispositivo puede estar conectado o insertado dentro de una entidad física o monitorear una entidad física en su entorno (Minerva et al., 2015).

2.2.2 Actuador

Un actuador es un dispositivo mecánico para el movimiento o control de un mecanismo o sistema. Este recibe energía, generalmente transportada por aire, corriente eléctrica o líquida y convierte esto en estados cambiantes, los cuales afectan a una o más entidades físicas (Minerva et al., 2015).

2.2.3 Aplicaciones

Una aplicación es un artefacto de software que implementa una lógica de negocio. Las aplicaciones acceden a recursos necesarios para lograr el objetivo de la lógica de negocio a través de servicios. Las aplicaciones pueden ser por ejemplo implantadas en un dispositivo, en un sistema empresarial o en



la nube. Una aplicación de un dispositivo es dependiente del hardware (Minerva et al., 2015).

2.2.4 Recursos

Un recurso es un elemento computacional, que permite el acceso a la información mediante capas con capacidades de actuación o mediante una entidad física (Minerva et al., 2015).

2.2.5 Servicios

Un servicio es un componente de software que habilita la interacción con recursos a través de una interfaz bien definida, a menudo vía Internet. Estos pueden ser combinados con servicios no necesariamente IoT (Minerva et al., 2015).

2.2.6 Características fundamentales

Las características fundamentales de IoT son las siguientes:

Característica	Descripción
Interconectividad de las cosas	Esto se refiere a que un sistema negocia con la interconexión de las cosas. Cada cosa es un objeto físico que es relevante para un usuario o relevante desde la perspectiva de la aplicación.
Conexión de las cosas a internet	Partiendo de la definición de IoT, se puede decir que cada “cosa” está conectada a internet. Por lo tanto se puede deducir que el sistema no es intranet o extranet de las cosas.
Identificación única	Un sistema IoT está compuesto por objetos que tienen una identificación única.
Ubicuidad	Según la definición de la ITU (ITU, 2005) la ubicuidad es la característica más importante en un sistema IoT, lo que indica que una red está disponible donde sea y cuando sea. Pero en el contexto de IoT donde sea y cuando sea se refiere a globalmente y siempre respectivamente.
Capacidad de sensado y actuación	Los sensores y actuadores están conectados a las “cosas” y realizan el procesamiento de esos datos que dan “inteligencia” a las cosas.
Inteligencia embebida	Los objetos inteligentes y dinámicos con un comportamiento emergente, inteligencia embebida y con funciones de conocimiento sirven como herramientas y llegan a ser

	extensiones del cuerpo y mente del ser humano.
Capacidad de comunicación interoperable	Un sistema IoT tiene la capacidad de comunicación basada en estándares de comunicación y protocolos interoperables.
Auto-configuración	La autoconfiguración consiste primordialmente en el descubrimiento de sus vecinos mediante servicios, organización de la red y aprovisionamiento de recursos (Chatzigiannakis et al., 2012). La dirección natural de los dispositivos IoT es el manejo y configuración de su software y hardware y la utilización de sus recursos (energía, comunicación, ancho de banda, acceso, etc.)
Cosas Programables	Las cosas en un sistema IoT tienen la característica de ser programables. A un nivel simple, un dispositivo programable puede tomar diferentes comportamientos mediante comandos de un usuario sin ser requeridos cambios físicos.

Tabla 2-3 Características de IoT. Fuente: IEEE-Internet of Things (Minerva et al., 2015).

2.2.7 Modelo de referencia de IoT

De acuerdo a CISCO (Green, 2014), el modelo consta de siete capas, mostradas en la Figura 2-3, las cuales se comunican de forma bidireccional, éstas se describen en las subsecciones siguientes.

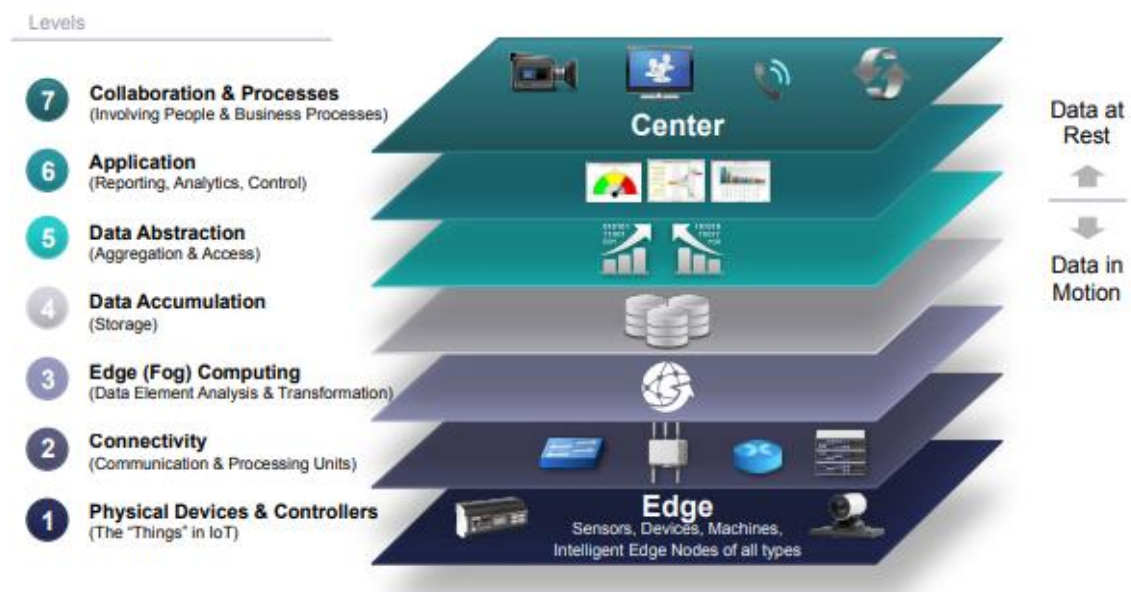


Figura 2-3 Modelo de Referencia IOT CISCO. Fuente: Green (2014)



2.2.7.1 Dispositivos físicos y controladores

En esta capa se pueden controlar múltiples dispositivos (cosas). Actualmente, la lista de dispositivos es extensa y cada día se agregan más. Además, los dispositivos son diversos, por lo que no existen reglas acerca de su tamaño, localización, forma u origen.

Los dispositivos son capaces de:

- Conversión análoga a digital si se requiere.
- Generar datos e información.
- Ser consultados y controlados en la red.

2.2.7.2 Conectividad

La función más importante es la transmisión de información confiable y sincronizada. La transmisión de información se da:

- Entre dispositivos (Capa 1)
- A través de redes (horizontalmente)
- Entre redes (Capa 2) y procesamiento de información a bajo nivel que ocurre en la Capa 3.

Uno de los objetivos principales del modelo de referencia IoT es la comunicación y procesamiento en redes de comunicación existentes; por lo que este modelo de referencia, no requiere una creación de una red diferente. Sin embargo, algunos dispositivos no tienen una dirección IP, por lo que es necesario introducir pasarelas de comunicación. La comunicación incluye:

- Comunicación con y entre dispositivos en la Capa 1.
- Implementación de varios protocolos.
- Conmutar y enrutar (Switching and Routing)
- Traducción entre protocolos.
- Seguridad a nivel de red.
- Análisis de red.

2.2.7.3 Edge (Fog) Computing

La función de esta capa se encarga de convertir flujos de datos de red a información adecuada para el procesamiento en la Capa 4. Esto significa que en esta capa se realiza la transformación y el análisis de un gran volumen de datos. En esta capa no existen transacciones o sesiones, sino solamente su concepto simple de unidades de datos, por lo que el procesamiento es limitado. La Capa 3 incluye:

- Filtrado de datos, limpieza y agregación.
- Inspección del contenido de los paquetes.
- Combinación del análisis a nivel de datos y red.
- Generación de eventos.

2.2.7.4 Almacenamiento de datos

Debido a que los datos están continuamente en movimiento, son convertidos a un formato de transferencia. La Capa 4 determina:

- Si los datos deben persistir; es decir determina si es necesario guardarlos permanentemente o en memoria volátil.
- El tipo de almacenamiento necesario. La persistencia puede requerir un sistema de archivos, un sistema de big data, una base de datos relacional, etc.
- Si los datos están organizados correctamente para el tipo de almacenamiento requerido.
- Si los datos deben ser combinados o procesados nuevamente con otros; por ejemplo, con datos recopilados anteriormente.

2.2.7.5 Abstracción de Datos

Los sistemas IoT podrían ser escalados a niveles corporativos o globales; para esto se requiere sistemas que permitan almacenar datos en otros sistemas como ERP, HRMS, CRM, entre otros. Las funciones que se realizan en esta capa, están enfocadas en renderizar los datos, de manera que se adapten de manera simple a otras aplicaciones.

Debido a que existen múltiples dispositivos generando datos, algunas razones por las que no se deben almacenar datos en un mismo lugar a la vez se describen a continuación:

- Existen demasiados datos como para guardarlos en un solo lugar.
- Enviar datos a una base de datos puede consumir mucho poder de procesamiento. En este aspecto existen soluciones como bases de datos OTPL (online transaction processing) y data warehouse.
- Los dispositivos podrían estar separados geográficamente, por lo que es óptimo procesarlos localmente.
- Pueden ser requeridos diferentes tipos de procesamiento.

2.2.7.6 Aplicación

En esta capa se interpreta la información; aquí, el software interactúa con el nivel 5 y los datos como REST. En este modelo de referencia esta etapa no se define estrictamente como una aplicación, sino que ésta se puede basar en



mercados verticales, naturaleza de los datos del dispositivo y necesidades de negocio. Aplicaciones de monitoreo y control representan diferentes modelos de aplicaciones, patrones de programación y pilas de software. Si en las Capas 1-5 están definidas e implementadas correctamente, el trabajo requerido en la Capa 6 se reduce.

2.2.7.7 Colaboración y procesos

Un sistema de IoT incluye procesos y personas, por lo que las personas son las que utilizan las aplicaciones asociadas a algunos datos para cumplir con necesidades específicas. Las aplicaciones permiten entregar datos correctos, en el tiempo correcto para que las personas realicen las cosas correctas. Sin embargo, la mayoría de sistemas requieren más de una persona, por lo que la colaboración y la comunicación es necesaria. El objetivo de esta capa es relacionar a personas y procesos de negocio.

2.2.8 Protocolos para comunicación

2.7.8.1 HTTP

El Protocolo de transferencia de hipertexto (HTTP) es un protocolo de nivel de aplicación para sistemas de información hipermedia distribuidos, colaborativos. Es un protocolo genérico, sin estado, que se puede utilizar para muchas tareas más allá de su uso para hipertexto, como servidores de nombres y sistemas de administración de objetos distribuidos, mediante la extensión de sus métodos de solicitud, códigos de error y encabezados. Una característica de HTTP es el tipado y la negociación de la representación de datos, permitiendo que los sistemas se construyan independientemente de los datos que se transfieren (Leach et al., 1999).

2.7.8.2 REST

El estilo de transferencia de estado representacional (*Representational State Transfer* - REST) es una abstracción de los elementos arquitectónicos dentro de un sistema hipermedia distribuido. REST ignora los detalles de la implementación del componente y la sintaxis del protocolo para centrarse en los roles de los componentes, las restricciones sobre su interacción con otros componentes y su interpretación de elementos de datos significativos. Abarca las limitaciones fundamentales sobre los componentes, conectores y datos que definen la base de la arquitectura web y, por lo tanto, la esencia de su comportamiento como una aplicación basada en red. (Masse, 2011)

2.7.8.3 MQTT (Messaging Queuing Telemetry Transport)

Es un protocolo de mensajería de publicación / suscripción, extremadamente simple y ligero, diseñado para dispositivos con restricciones y redes de bajo ancho de banda, alta latencia o poco confiables. Los principios de diseño son minimizar el ancho de banda de la red y los requisitos de recursos del dispositivo al tiempo que se intenta garantizar la fiabilidad y cierto grado de seguridad de la entrega. Estos principios también hacen que el protocolo sea ideal para el mundo emergente de "máquina a máquina" (*Machine to Machine - M2M*) o "Internet de las cosas" de dispositivos conectados, y para aplicaciones móviles donde el ancho de banda y la potencia de la batería son de primera. (MQTT.org, 2017)

2.2.8 Estándares

Internet of Medical Things (IoMT) comprende una red de dispositivos conectados que detecta datos vitales en tiempo real y permite la interacción de máquina a máquina y soluciones de intervención en tiempo real que transformarán radicalmente la entrega de servicios de salud, la asequibilidad y la confiabilidad en un futuro cercano. Además, una mayor participación del paciente en la toma de decisiones impulsará el cumplimiento del servicio de atención médica. Como resultado de esto, la tasa de adopción de tecnología aumentará en los próximos años para hacer crecer este mercado y alcanzar los \$ 156 mil millones para 2020 (Shelke & Sharma, 2016).

Es por ello que es necesario adoptar y orientar el desarrollo de dispositivos a cumplir estándares médicos que permitan hacerlos confiables a médicos y pacientes. Zhao, Wang y Nakahira (2011) investigan los estándares de salud de cómo acceder y mostrar los datos compartidos por las organizaciones. También estudian los diseños de una interfaz entre una plataforma y un establecimiento médico y la construcción de la red; es por ello que utilizan estándares como: HL7 / XML, DICOM (comunicación digital por imágenes), ADSL, WLAN, 3G y WCDMA.

2.3 Integración de Computación en la Nube e IoT (CloudIoT)

La Computación en la Nube e IoT son tecnologías que han evolucionado de manera independiente, en la literatura se han identificado ventajas de su integración (Botta et al., 2014; Babu, Lakshmi y Rao, 2015). IoT se ve beneficiado de los recursos virtualmente ilimitados de la Nube, en términos de poder de almacenamiento y procesamiento. Por otro lado, la Nube extiende su alcance para interactuar con cosas del mundo real de una manera más distribuida y dinámica, y ofrece los servicios de IoT en un enfoque pago por uso. La Nube actúa como capa intermedia entre las cosas y las

aplicaciones/usuarios, ocultando así la complejidad de la Nube. En la Tabla 2-4, las características de la integración entre la Computación en la Nube e IoT son resumidas.

IoT	Computación en la Nube
Cosas del mundo real	Recursos virtuales
Capacidades computacionales limitadas.	Capacidades computacionales virtualmente ilimitadas
Almacenamiento limitado o sin capacidades de almacenamiento	Capacidades de almacenamiento virtualmente ilimitadas
Internet como punto de convergencia	Internet para la entrega del servicio
Gran fuente de datos	Medios para administrar grandes cantidades de datos

Tabla 2-4 Características complementarias Computación en la Nube e IoT. Fuente: Botta et al. (2014).

Los factores que impulsan la integración de la Computación en la Nube y IoT se resumen en la tabla 2-5. (Botta et al., 2016)

Factor	Descripción
Almacenamiento	IoT implica por definición, a una gran cantidad de fuentes de información (es decir, las cosas), que producen una gran cantidad de datos no estructurados o semiestructurados (European Commission, 2013). El almacenamiento a gran escala y de larga duración es posible gracias a la capacidad de almacenamiento prácticamente ilimitada, de bajo costo y bajo demanda provista por la Nube.
Computacional	Los datos recopilados por las cosas generalmente se transmiten a nodos más potentes donde el procesamiento es posible. La nube ofrece capacidades de procesamiento prácticamente ilimitadas y un modelo de uso bajo demanda. Las necesidades de procesamiento de IoT pueden satisfacerse adecuadamente para realizar análisis de datos en tiempo real, para gestionar eventos complejos (B.B, Saluia, Sharma, Mittal, y Sharma, 2012), para implementar aplicaciones escalables, colaborativas, centradas en sensores y para apoyar la descarga de tareas para el ahorro de energía.
Comunicación	Las aplicaciones ubicuas personalizadas se pueden entregar a través de IoT, mientras que la automatización se puede aplicar a la recopilación y distribución de datos a bajo costo.

	La computación en la nube es una solución efectiva y económica para conectar, rastrear y administrar cualquier cosa desde cualquier lugar y en cualquier momento mediante el uso de portales personalizados y aplicaciones integradas (B.B et al., 2012).
Alcance	La adopción del paradigma CloudIoT permite nuevos servicios inteligentes y aplicaciones que permiten a la nube lidiar con una serie de escenarios nuevos de la vida real.

Tabla 2-5 Factores para la integración CloudIoT. Fuente: Botta et al. (2016).

2.3.1 Aplicaciones

2.3.1.1 Cuidado de la salud

Nuevas soluciones para las tecnologías de información en el campo médico, en las áreas de ambiente de vida asistido (Ambient Assisted Living - AAL) y telemedicina han emergido (Zhang et al., 2010; Dash, Mohapatra y Pattnaik, 2010; Kuo, 2011; Doukas y Maglogiannis, 2012). Los dispositivos inteligentes contribuyen a la innovación en la salud, ofreciendo servicios médicos ubicuos de alta calidad. Las aplicaciones médicas generan una gran cantidad de información relacionada con el paciente y su estilo de vida, siendo almacenada para análisis y procesamiento en la Nube. Además, CloudIoT permite que los dispositivos móviles sean adecuados para la entrega de información de salud, acceso y comunicación, mejorando la seguridad de los datos médicos, la disponibilidad y redundancia. (Botta et al., 2014)

Los ambientes de vida asistidos, son parte de la inteligencia ambiental, representan un nuevo paradigma en tecnología de la información, destinada a potenciar las capacidades de las personas través de entornos digitales que son sensibles, adaptables y receptivos a las necesidades humanas. (Fariba Sadri, 2011).

El objetivo de los ambientes de vida asistidos es explotar los potenciales proporcionados por las nuevas tecnologías de información y comunicación para ayudar a las personas mayores y en recuperación a mantener un estilo de vida activo, independiente y seguro en su rutina diaria. AAL puede usarse para prevenir y curar enfermedades, mejorar el bienestar y las condiciones de salud de los adultos mayores. Las herramientas de AAL, como las herramientas de administración de medicamentos y los recordatorios de medicamentos, permiten a los adultos mayores controlar sus condiciones de salud. Las tecnologías AAL también pueden proporcionar más seguridad para los adultos mayores, utilizando sistemas móviles de respuesta de emergencia, sistemas de detección de caídas y sistemas de videovigilancia. Otras tecnologías de AAL

brindan ayuda con las actividades diarias, basadas en el monitoreo de las actividades de la vida diaria (AVD, Activities of Daily Living) y la emisión de recordatorios, además de ayudar con la movilidad y la automatización. Tales tecnologías pueden permitir que los adultos mayores se conecten y se comuniquen mejor con sus pares, así como con sus familiares y amigos. (Rashidi y Mihailidis, 2013)

2.3.1.2 Ciudades inteligentes (Smart Cities)

CloudIoT conduce a la generación de servicios que interactúan con el entorno, el desarrollo sostenible de las áreas urbanas es un desafío de importancia clave y requiere tecnologías y servicios nuevos, eficientes y fáciles de usar. El desafío es aprovechar el poder de colaboración de las redes de TIC (redes de personas, de conocimiento, de sensores) para crear una conciencia colectiva e individual sobre las múltiples amenazas de sostenibilidad a las que se enfrenta nuestra sociedad hoy en día a nivel social, ambiental y político (Botta et al., 2014). CloudIoT puede proporcionar un middleware común para servicios de ciudad inteligente orientados al futuro (Suciu et al., 2013; Zanella et al., 2014), adquiriendo información de diferentes infraestructuras de detección heterogéneas, accediendo a todo tipo de tecnologías de geolocalización e IoT, y exponer información de manera uniforme.

2.3.1.2.1 Hogar inteligente y medición inteligente

CloudIoT tiene una gran aplicación en entornos domésticos, en donde la adopción conjunta de dispositivos integrados heterogéneos y la Computación en la Nube permite la automatización de actividades internas comunes. Varias aplicaciones de hogar inteligente propuestas en literatura involucran redes de sensores (inalámbricas) y realizan la conexión de dispositivos inteligentes a Internet para monitorear remotamente su comportamiento o controlar de forma remota (p. ej. para gestionar la iluminación, la calefacción y el aire acondicionado). (Babu et al., 2015)

2.3.1.2.2 Video vigilancia

La videovigilancia es importante como parte del problema relacionado con la seguridad, ya que funciona como un sistema de monitoreo y autogestión. El complejo análisis de video requiere soluciones basadas en la nube (Prati, Vezzani, Fornaciari y Cucchiara, 2013), para satisfacer los requisitos de almacenamiento y procesamiento. La solución propone, identificar el almacenamiento y la administración de la información de video desde la cámara y la eficiencia de la entrega de datos hasta la cantidad de usuarios a través de internet, equilibrio de carga y tolerancia a fallas.



2.3.1.3 Movilidad inteligente

IoT ofrece una solución prometedora para los servicios de automóviles y los sistemas de transporte. Las tecnologías de integración de computación en la nube con redes satelitales, redes de sensores inalámbricos representan una oportunidad para enfrentar los principales desafíos actuales. IoT basado en datos vehiculares es una nueva generación de tecnología que se puede implementar para brindar beneficios comerciales, como la reducción de la congestión vial, la gestión del tráfico (Wu He, Gongjun Yan y Li Da Xu, 2014), el aumento de la seguridad vial.

2.3.1.4 Energía inteligente y red inteligente

IoT y la Computación en la Nube se pueden fusionar eficazmente para proporcionar una gestión inteligente de la distribución y el consumo de energía en entornos heterogéneos (Botta et al., 2014). La adopción de la nube conduce a aumentar la confiabilidad al proporcionar participación de los usuarios, para lograr generación distribuida, calidad de la electricidad y respuesta a la demanda (Miao Yun y Bu Yuxin, 2010). La computación en la nube hace posible analizar y procesar grandes cantidades de datos e información procedente de diferentes fuentes distribuidas a lo largo de redes de área amplia, con el fin de implementar un control inteligente de los objetos.

2.4 Lenguajes de dominio específico (*Domain Specific Language - DSL*)

Los DSLs son lenguajes adaptados a un dominio de aplicación específico que ofrecen poder expresivo centrado en un dominio de problema particular. La contribución principal de las DSL es permitir la reutilización de artefactos de software. Entre los tipos de artefactos que pueden reutilizarse a través de DSL se encuentran el código fuente, los diseños de software y las abstracciones de dominio.

2.5 Ingeniería Dirigida por Modelos (*Model Driven Engineering - MDE*)

Model Driven Engineering (MDE) proporciona un mayor nivel de abstracción con respecto a la ingeniería de software. De forma análoga al principio de que todo es un objeto en el paradigma orientado a objetos, MDE adopta el principio de que todo es un modelo (Bézivin, 2004). Un modelo representa un sistema (parte de la realidad) y se expresa en un lenguaje de modelado. En MDE, la estructura del lenguaje de modelado está dada por otro modelo, llamado metamodelo (Wolterink, 2009).

Los modelos y metamodelos se pueden colocar en capas, según la convención

MDE, la realidad está en la capa M0, los modelos que representan la realidad están en la capa M1 y los metamodelos de esos modelos están en la capa M2. Finalmente, los metametamodelos de esos están en M3. En la Figura 2.5 se observa la arquitectura en capas de MDE. Como un metamodelo es un modelo, se expresa en un lenguaje de modelado. El modelo de este lenguaje de modelado es el metamodelo. Esto sugiere una iteración infinita, para limitar la iteración de la relación de conformidades. Se dice que un modelo de metamodelos se conforma a sí mismo. A continuación se dan algunas definiciones informales inspiradas en las definiciones analizadas por Kurtev (Kurtev, 2005):

- **Definición 1 (Modelo)** Un modelo (en M1) es una abstracción de una parte de la realidad para un propósito específico. Un modelo se expresa en un lenguaje de modelado.
- **Definición 2 (Lenguaje de modelado)** Un lenguaje de modelado es un lenguaje bien entendido (no siempre formal) que describe los conceptos y su relación de una parte de la realidad.
- **Definición 3 (MetaModelo)** Un metamodelo (en M2) es un modelo de lenguaje de modelado.
- **Definición 4 (MetaMetaModelo)** Un metametamodelo (en M3) es un modelo de un lenguaje de modelado que se puede usar para expresar metamodelos. El modelo del lenguaje de modelado del metamodelo está expresado por sí mismo.

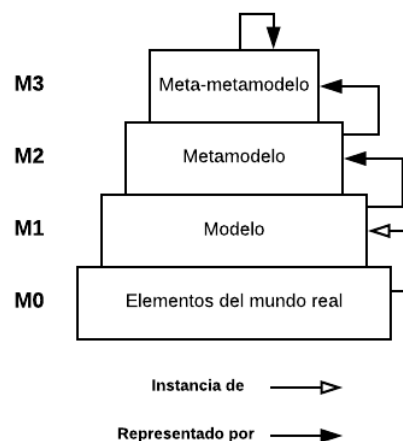


Figura 2-4 Arquitectura MDE. Fuente: Arquitectura Ingeniería Dirigida por Modelos (Kurtev, 2005).

2.6 Modelo de Dominio Específico (Domain Specific Model- DSM)

En MDE, los lenguajes de dominio específico pueden ser generados a partir de un metamodelo. El metamodelado es una técnica común para conceptualizar un dominio al definir la sintaxis abstracta y la semántica estática de un DSM.



Un metamodelo define un conjunto de elementos de modelado y sus relaciones válidas que representan ciertas propiedades para un dominio específico. Los modelos específicos del dominio se pueden crear utilizando este DSM y se pueden traducir a código fuente, o se pueden sintetizar en datos para enviarlos a una herramienta de simulación (Schmidt, 2006).

Los DSM se describen utilizando metamodelos, que definen las relaciones entre los conceptos en un dominio y especifican con precisión la semántica y las restricciones asociadas con estos conceptos de dominio. Al generar un modelo, los desarrolladores usan lenguajes DSM (DSML) para crear aplicaciones (Lin, Gray y Jouault, 2007).

Capítulo 3

Estado del Arte

En el presente capítulo se realiza una breve síntesis de las arquitecturas CloudIoT, que se refiere a las infraestructuras de alto nivel dirigidas a la integración de IoT y la Computación en la Nube y la definición de componentes y sus respectivos roles. Se analizan las principales diferencias entre ellas y cómo se han aplicado en diferentes áreas.

3.1 Integración Computación en la Nube e IoT (CloudIoT)

Aazam y Huh (2014) proponen una arquitectura basada en Fog Computing y una pasarela inteligente. Con esta arquitectura, parte del procesamiento se realiza en la pasarela, antes de que los datos se envíen a la nube; esto reduce los recursos que se utilizan en la nube. En este caso, una “pasarela inteligente” se encarga de filtrar los datos, mejorar el procesamiento, enviar sólo datos necesarios y reconstruirlos para ser enviados a la nube. Al integrar Fog Computing, que es una extensión de Computación en la nube, con la diferencia de que no todos los datos se almacenan en la nube, sino también en los dispositivos, se tiene una arquitectura de red inteligente que maneja los recursos en los dispositivos y en una pasarela, antes de ser enviados a la nube. En la figura 3-1, se observa la arquitectura que ha sido planteada y en la figura 3-2 se observa la arquitectura por capas.

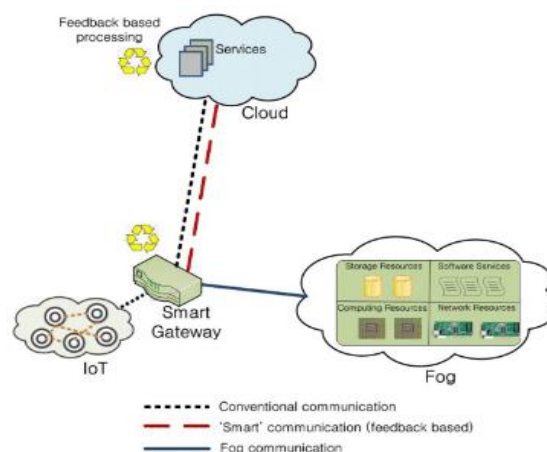


Figura 3-1 Pasarela Inteligente con Fog Computing. Fuente: Aazam y Huh (2014)

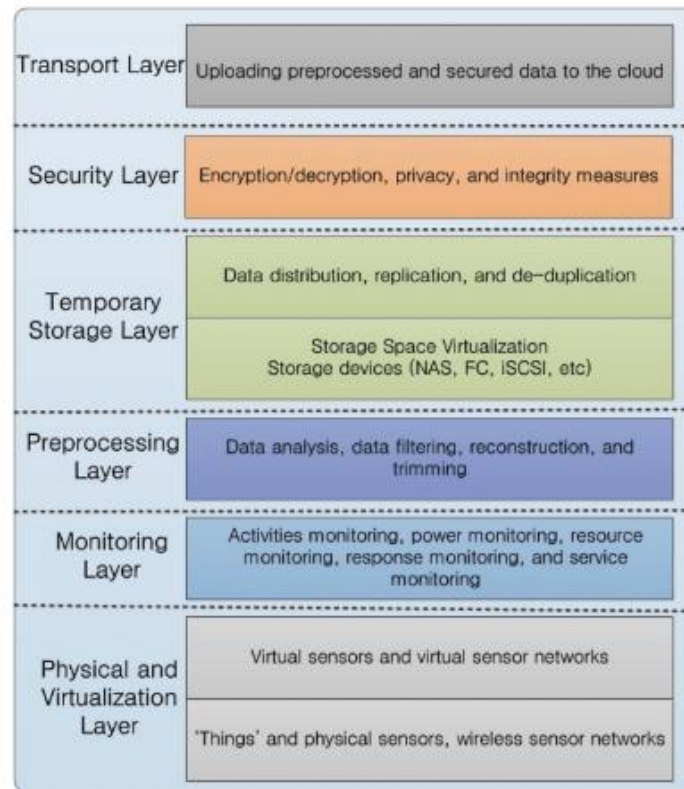


Figura 3-2 Arquitectura en Capas de Pasarela Inteligente con Fog Computing. Fuente: (Aazam y Huh, 2014).

Distefano, Merlino y Puliafito (2013) proponen un tipo de Infraestructura como servicio (Infrastructure as a Service – IaaS), en el contexto de “Nube de las Cosas” o “*Cloud of Things*”. Su enfoque se puede definir con el acrónimo SAaaS (*Sensing and Actuation as a Service*), es decir, detección y actuación como servicio. Contribuyen al diseño de una arquitectura en donde los servicios de nueva generación aprovechan el entorno, recolectando datos y aplicando estrategias de control.

En la parte inferior de la pila (ver Figura 3-3), el módulo SAaaS es necesario para la abstracción de los recursos de detección y actuación, mejorados con políticas y manejo de datos. Por lo tanto, el módulo SAaaS proporciona conjuntos de sensores / actuadores en el sentido más amplio posible (por ejemplo, impresoras, cámaras web, periféricos de interacción humana).

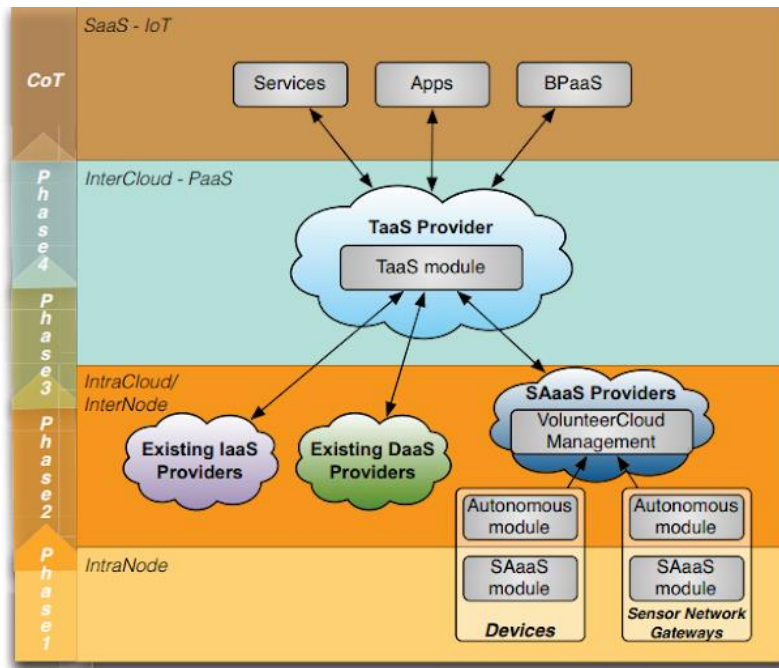


Figura 3-3 Esquema arquitectónico y módulos. Fuente: Distefano et al. (2013)

La arquitectura propuesta por Wu et al. (2012) se basa en el paradigma GaaS (Gateway as a Service) en donde la pasarela es el núcleo, éste comunicará los diferentes tipos de sensores y actuadores con Internet. Esto permite que los clientes puedan consumir servicios y construirlos en su propio software en diferentes niveles; además, los desarrolladores pueden construir su propio software utilizando APIs programables en la segunda capa. En la Figura 3-4 se observa la arquitectura basada en GaaS.

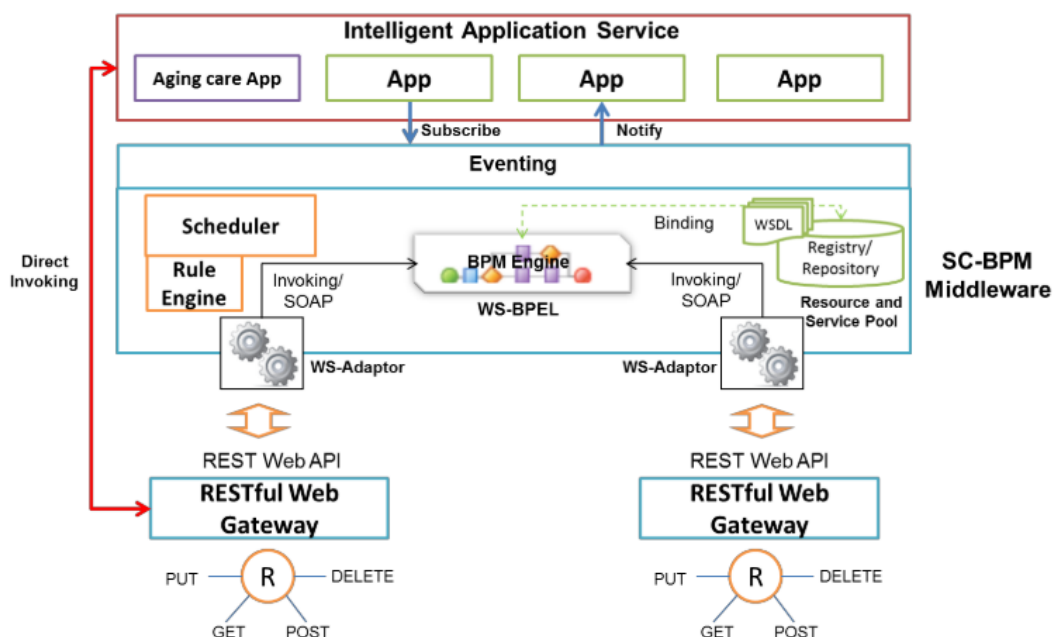


Figura 3-4 Arquitectura GaaS. Fuente: Wu et al. (2012)

En esta arquitectura, la parte más importante es el componente RESTful Web Gateway, ya que se encarga de la comunicación con los dispositivos físicos que no tienen acceso a internet y abstrae sus datos mediante APIs físicas programables. Entonces, la pasarela puede directamente proveer de servicios web atómicos de los dispositivos para otros consumidores de los servicios. En la Figura 3-5 se observa detalladamente el componente con sus características.

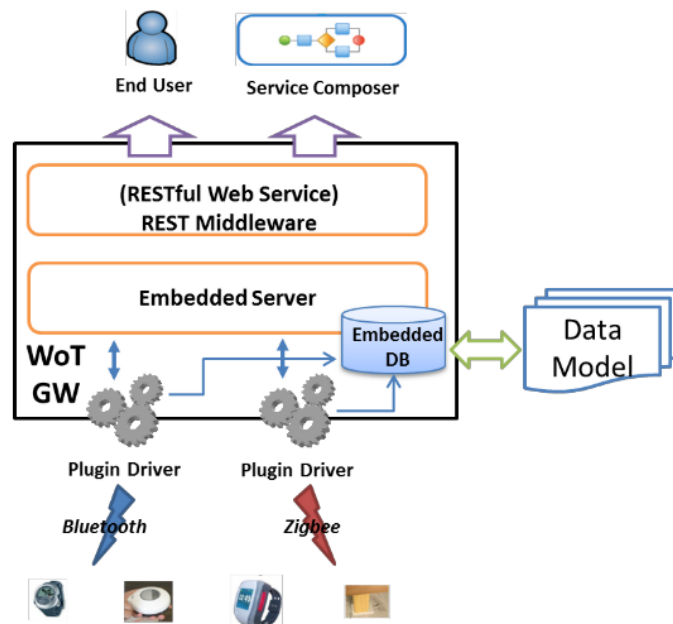


Figura 3-5 Pasarela Restful Web. Fuente: Wu et al. (2012)

En el trabajo de Hassan, Albakr y Al-dossari (2014) se propone un marco de IoT asistido por la nube, para abordar los problemas desafiantes en el dominio de cuidado de salud: transmisión confiable de datos de signos vitales a la nube, asignación dinámica de recursos para facilitar el acceso y procesamiento sin interrupciones de los datos de IoT y técnicas efectivas de minería de datos. La Figura 3-6 muestra la arquitectura del sistema de una plataforma general CloudIoT.

Para la adquisición de datos, la información contextual se recopila a partir de sensores portátiles y móviles. Los datos monitorizados recopilados se transmiten al dispositivo móvil a través de Bluetooth y luego al servidor de la nube a través de Internet o 3G. Servicios web REST son usados para enviar datos de sensores y recuperar información de la nube.

Los servidores en la nube proporcionan potentes recursos de máquinas virtuales (VM) bajo demanda, la administración de estos datos de forma ubicua se realiza mediante una variedad de interfaces como computadora personal, TV y teléfonos móviles. El sistema integrado propuesto proporciona diversos

servicios IoT, como pre procesamiento en tiempo real, almacenamiento, intercambio, priorización, visualización, análisis, resumen y búsqueda de datos monitoreados. La característica principal de la arquitectura propuesta es que puede facilitar la creación de una plataforma de servicios de atención de salud inteligente (es decir, capacidad de detección sensible al contexto).

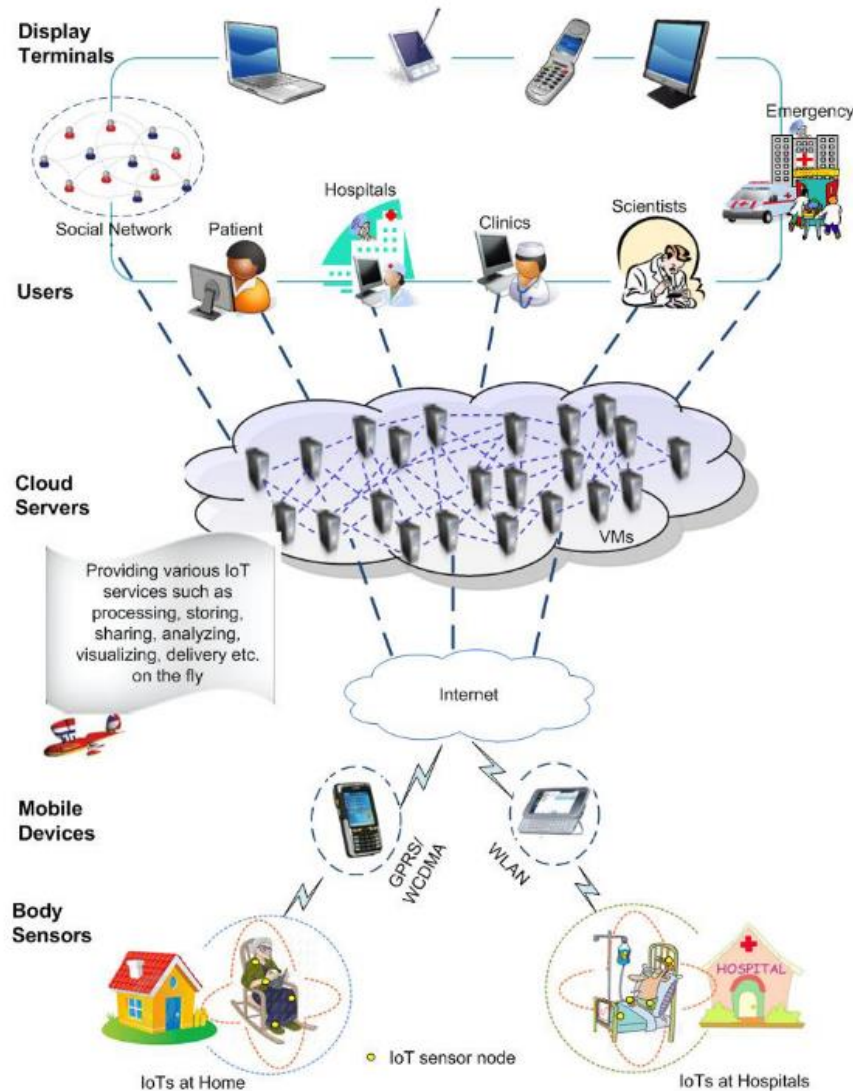


Figura 3-6 Marco general propuesto para la nube de IoT en el escenario de cuidado de la salud.

Fuente: Hassan et al. (2014)

En el estudio presentado por Puliafito (2014) se propone una arquitectura que provee a los usuarios de Internet, la capacidad de obtener cualquier tipo de datos adquiridos de diferentes infraestructuras heterogéneas y que son expuestos de manera uniforme. El sistema integrado de sensores en el entorno de la nube para la gestión avanzada de riesgos múltiples (SIGMA) es una arquitectura basada en capas cuya función es adquirir, integrar y calcular datos heterogéneos de diferentes sensores de la red (clima, sismos, volcanes, tráfico,

etc.). Este sistema es utilizado para que en caso de que ocurra una situación de riesgo, el sistema responda ante ellos con planes de respuesta y con desarrollo e integración de soluciones. Esta arquitectura está compuesta por seis capas descritas en la Figura 3-7.

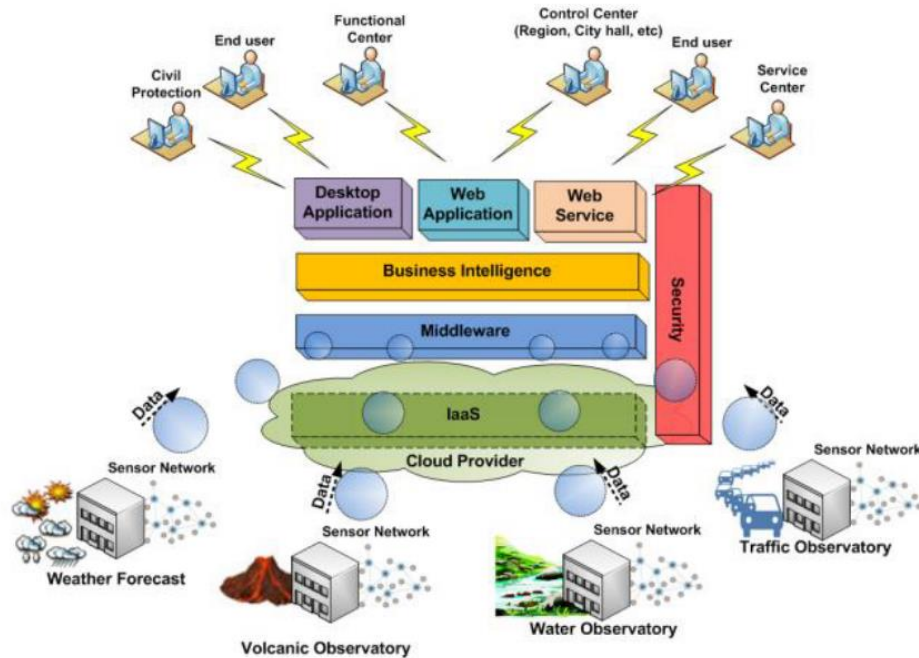


Figura 3-7 Arquitectura proyecto SIGMA. Fuente: Puliafito

Al nivel más bajo, se pueden incluir diferentes redes de sensores, que se encargan de monitorear aspectos importantes de la ciudad. En la figura 3-8 se puede observar cómo esta arquitectura se puede integrar con los datos que han sido tomados de las diferentes estaciones.

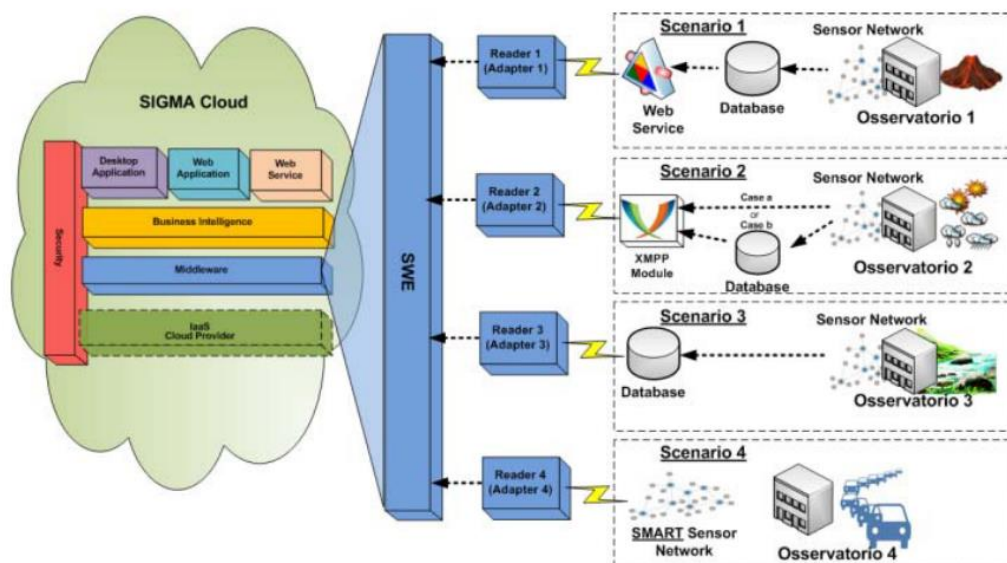


Figura 3-8 Escenario de Integración. Fuente: Puliafito (2014)

En el trabajo propuesto por Zhou et al. (2013), los autores proponen la arquitectura CloudThings, una plataforma de IoT basada en la nube. La figura 3-9 especifica las soluciones técnicas para crear redes de cosas, interactuar con cosas e integrar cosas con la nube.

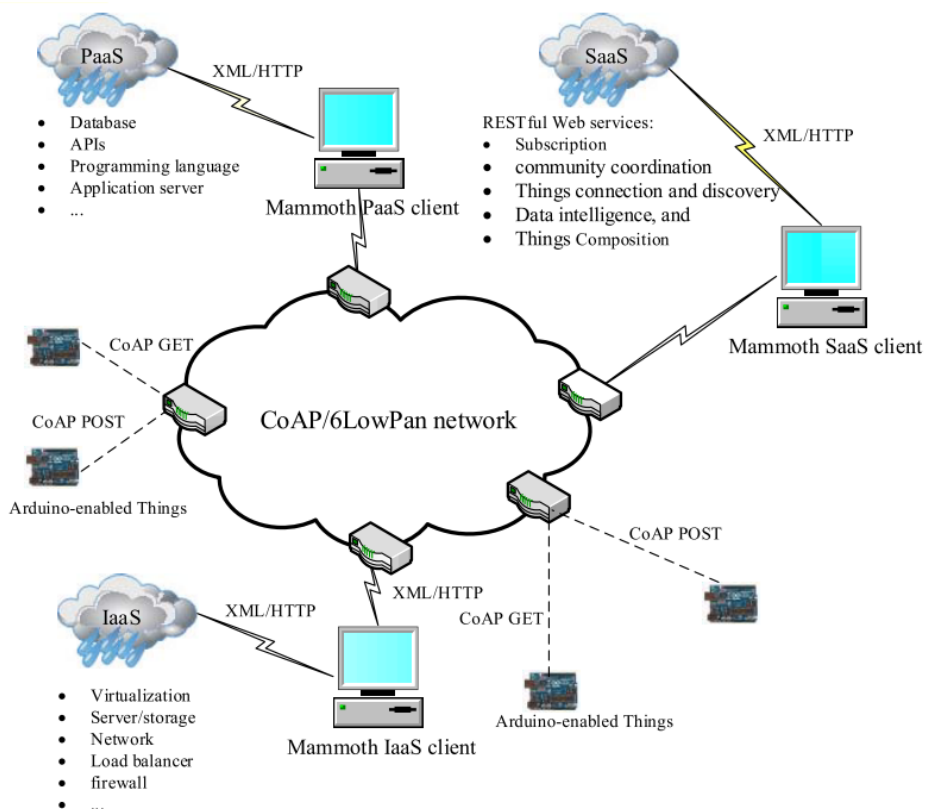


Figura 3-9 Arquitectura CloudThings: Plataforma IoT basada en la nube. Fuente: (Zhou et al., 2013)

Esta arquitectura se representa como una plataforma en línea que permite a los integradores de sistemas y proveedores de soluciones aprovechar una infraestructura completa de aplicaciones IoT para desarrollar, implementar, operar y componer aplicaciones y servicios que constan de tres módulos principales:

- La plataforma de servicios IoT es un conjunto de servicios en la nube (IaaS) que permite a los usuarios ejecutar cualquier aplicación en el hardware de la nube.
- La suite de desarrolladores para IoT es un conjunto de herramientas de servicio en la nube (PaaS) para el desarrollo de aplicaciones IoT. Estas herramientas incluyen interfaces de programación de aplicaciones (API) de servicios web abiertos, que proporcionan capacidades completas de desarrollo e implementación a los desarrolladores.

- El portal de operaciones para IoT es un conjunto de servicios en la nube (SaaS) que admiten la implementación y manejan o respaldan servicios especializados de procesamiento.

3.1.1 Cuidado de la Salud (*Healthcare*)

Doukas y Maglogiannis (2012) presentan una arquitectura orientada a dispositivos médicos que aprovecha las características de la Nube e IoT para enviar datos en tiempo real de manera eficiente. La arquitectura está compuesta por cinco partes descritas a continuación:

1. *Wearables* y dispositivos con sensores que leen las bio-señales del paciente, movimiento e información médica.
2. Un sensor que hace el papel de pasarela y captura todas las señales de los sensores para enviarlos hacia Internet. Se puede usar como pasarela un dispositivo móvil o un microcontrolador que permita una conexión a Internet. Además, la pasarela se encarga de capturar y analizar datos obtenidos de los sensores (cantidad de energía, forma de operación, etc.).
3. La API para comunicación provista por la plataforma en la nube; aquí, se utilizan interfaces livianas de comunicación (como servicios web REST) para que la pasarela pueda enviar y recibir información. La API también puede ser utilizada por aplicaciones externas para procesar información, generar alertas, generar pagos, etc.
4. Una interfaz web que se encarga de manejar la aplicación que es actualizada en tiempo real y proveer una vista de los datos de los sensores e información médica del paciente (localización, estado del paciente, etc.).
5. La infraestructura en la nube que aloja las interfaces y maneja la aplicación. Aquí se encuentran los recursos esenciales (como CPU, almacenamiento y servidores de aplicaciones) para desplegar las aplicaciones web y las interfaces que habilitan la comunicación con sensores y sistemas externos.

Las ventajas más importantes de esta arquitectura son la escalabilidad, interoperabilidad y acceso liviano. Además, en la comunicación de cada dispositivo con la nube es posible integrar características de seguridad como autenticación y encriptación de los datos. En la figura 3-10 se observa la arquitectura descrita.



Figura 3-10 Arquitectura de Computación en la Nube e IoT orientada a dispositivos médicos.

Fuente: Doukas y Maglogiannis (2012)

Una plataforma distribuida basada en Computación en la Nube para la gestión de datos de atención médica generalizada es presentada por Doukas, Pliakas, Tsanakas y Maglogiannis (2012). Este trabajo presenta un sistema integrado para la gestión de datos de sensores relacionados con la detección de caídas de personas con discapacidad o personas mayores.

La plataforma distribuida está basada en Cumulocity, una plataforma horizontal de Solución en la Nube Machine-to-Machine¹ proporcionada por Nokia Siemens Networks (NSN). La Figura 3-11 presenta una ilustración de la arquitectura propuesta para administrar datos de atención médica generalizados a través de la plataforma *Cumulocity Cloud*.

Se puede utilizar una variedad de sensores para controlar el estado y el contexto del paciente. Estos últimos pueden ser sensores portátiles y textiles que monitorean bioseñales vitales, el movimiento del paciente y generan alertas en casos de accidente cerebrovascular o detección de caídas. Todos los sensores están equipados con interfaces de red apropiadas (por ejemplo, WiFi, Bluetooth o ZigBee) para comunicarse directamente con la plataforma de la nube o a través de nodos intermedios, por ejemplo, como un teléfono inteligente. Las interfaces de software que pueden actuar como nodos intermedios para reenviar los datos a la nube utilizando llamadas al servicio web REST. La comunicación entre los sensores o los nodos intermedios y la nube se realiza a través del protocolo de seguridad *Security Sockets Layer* (SSL) que proporciona el cifrado esencial de los datos a través de la

¹ <https://www.cumulocity.com/>

transmisión. La plataforma Cumulocity se implementa en la parte superior de la infraestructura de Amazon AWS² y cumple con HIPAA³.

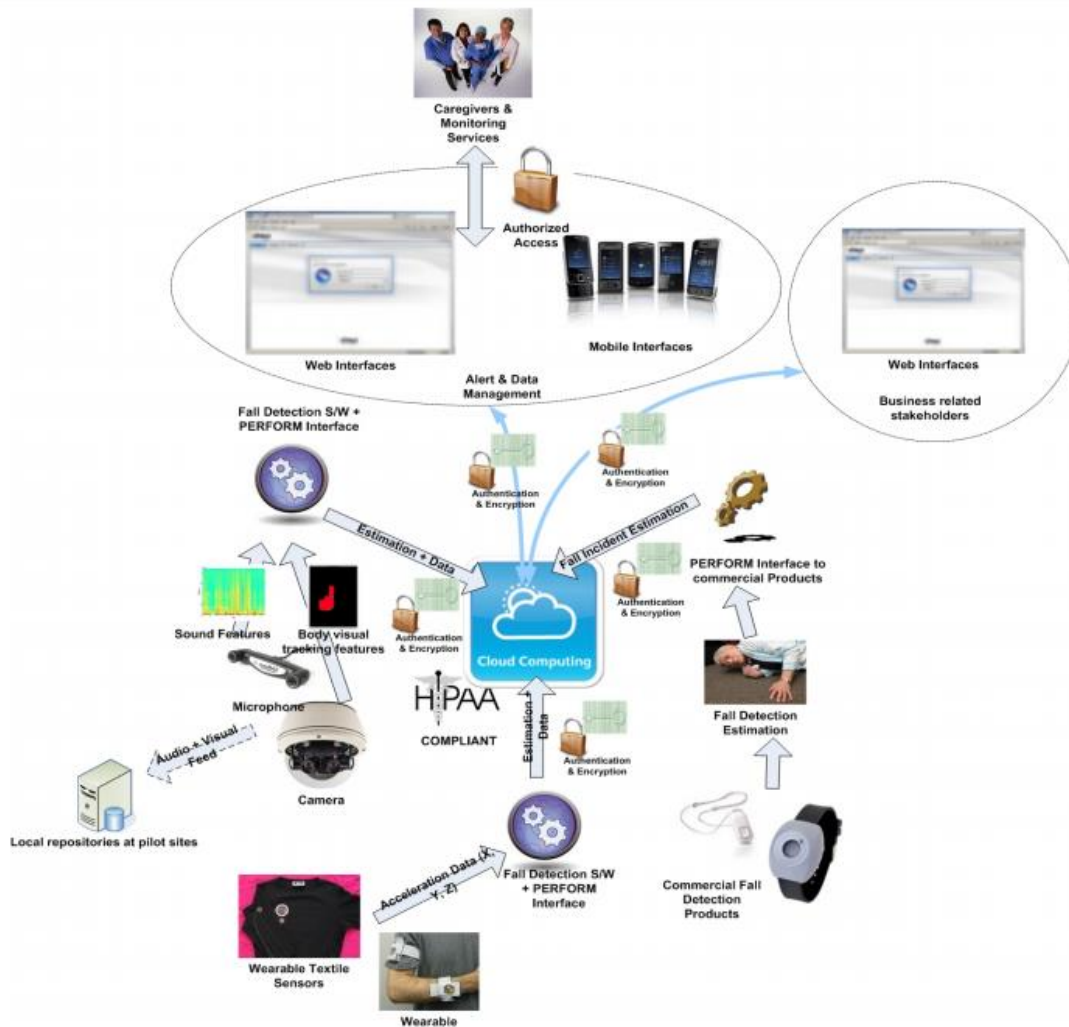


Figura 3-11 Arquitectura propuesta para gestionar los datos de atención médica generalizada en la nube. Fuente: Doukas et al. (2012)

El enfoque presentado por Mohammed et al. (2014) es construir una aplicación móvil basada en la plataforma Android para el dominio de la atención médica, que utiliza la idea de Internet of Things (IoT) y la computación en la nube. Una arquitectura diseñada en capas es propuesta. En este caso se dividen en tres capas importantes que son: capa de hardware, capa de aplicación y capa en la nube. A su vez se tienen 4 subcapas que son: Capa de servicio, capa de plataforma de aplicación y la capa de transferencia de archivos y escritura. En la figura 3-12 se puede observar la arquitectura en capas.

² <https://aws.amazon.com/es/>

³ <http://www.dhcs.ca.gov/formsandpubs/laws/hipaa/Pages/1.00WhatIsHIPAA.aspx>

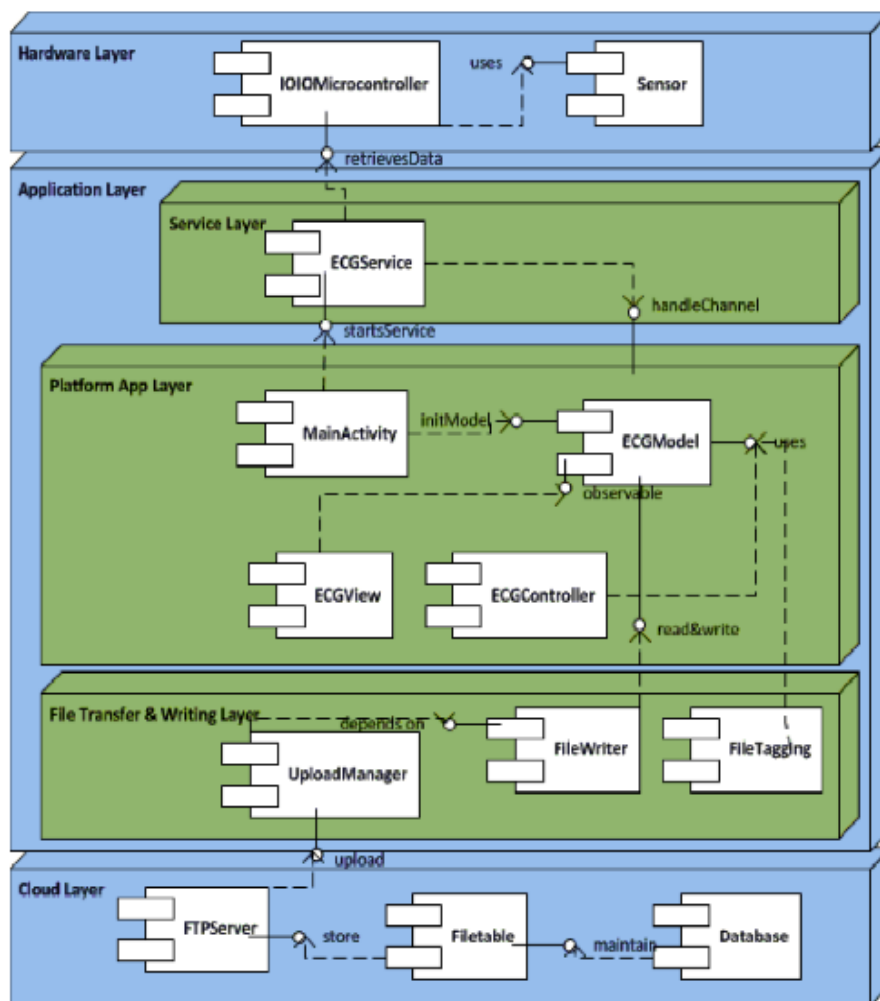


Figura 3-12 Arquitectura en capas propuesta. Fuente: Mohammed et al. (2014)

En la capa de aplicación un servicio para electrocardiogramas (ECGService) está corriendo por detrás de la aplicación, el cual es iniciado por la capa de aplicación por el MainActivity. El MainActivity funciona en un dispositivo Android y se comunica con la vista y el controlador de la capa de plataforma de la aplicación; de esta forma la transferencia y escritura de archivos se da a través del Tagging, el cual se comunica directamente con el modelo para la vista. El objetivo de esta arquitectura es tener buffers de memoria en cada capa para optimizar el proceso de comunicación entre capas y con los sensores.

En la tabla 3-1 se presenta en resumen de los trabajos de investigación analizados en esta sección, junto con elementos que aparecen en las arquitecturas de cada trabajo.

Paper	Gateway	Computación en la Nube	IoT	Fog	Componente Adaptable	Middleware	Integración sistemas externos	Módulos
Fog Computing and Smart Gateway Based Communication for Cloud of Things (Aazam & Huh, 2014)	X	X	X	X				
Towards the Cloud of Things Sensing and Actuation as a Service, a key enabler for a new Cloud paradigm (Distefano et al., 2013)	X	X	X		X			X
Gateway as a Service: A Cloud Computing framework for Web of Things (Wu et al., 2012)	X	X	X		X	X		X
A cloud-assisted Internet of Things framework for pervasive healthcare in smart city environment (Hassan et al., 2014)	X	X	X				X	
SensorCloud: An Integrated System for Advanced Multi-risk Management (Puliafito, 2014)	X	X	X		X	X	X	
CloudThings: a Common Architecture for Integrating the Internet of Things with Cloud Computing (Zhou et al., 2013)	X	X	X				X	
Bringing IoT and Cloud Computing towards Pervasive Healthcare (Doukas & Maglogiannis, 2012)	X	X	X					X
Distributed management of pervasive healthcare data through Cloud Computing (Doukas et al., 2012)	X	X	X					X
Internet of Things: Remote Patient Monitoring Using Web Services and Cloud Computing (Mohammed et al., 2014)	X	X	X					X

Tabla 3-1 Tabla comparativa revisión estado del arte. Fuente: Elaboración propia.

Cada “X” representa la presencia de los componentes correspondientes en la arquitectura correspondiente.

3.2 Discusión

Como se describe en el presente capítulo, la arquitectura de Aazam y Huh (2014) propone una solución mediante Fog Computing, el estudio realizado por Wu et al. (2012) presenta una arquitectura utilizando Gateway como Servicio, pero en ambos casos, no consideran la integración de sistemas externos. Distefano, Merlino y Puliafito (2013), presentan una arquitectura basada en infraestructura como servicio; sin embargo cada componente en la arquitectura es estrictamente necesario, lo que no la hace dinámica. En el trabajo de Hassan, Albakr y Al-dossari (2014), en el estudio presentado por Puliafito (2014) y , además, en el trabajo presentado por Zhou et al. (2013) se propone un marco de IoT asistido por Computación en la Nube, para abordar los problemas desafiantes en el dominio de cuidado de salud e integración de sistemas externos, pero no se define un Lenguaje de Dominio Específico que permita un fácil entendimiento para los expertos en el dominio de AAL y por desarrolladores. Adicionalmente, arquitecturas relacionadas a cuidado de la salud presentadas por Doukas y Maglogiannis (2012) y Doukas, Pliakas, Tsanakas y Maglogiannis (2012) agregan el uso de dispositivos médicos y *wereables*, sin embargo no analizan la integración con servicios y sistemas externos como hospitales o centros de cuidado. Por otra parte, la arquitectura propuesta por nosotros permite la integración de todos los componentes descritos en la Tabla 3-1, valiéndonos para este propósito del modelado y lenguaje de dominio específico.

Capítulo 4

Modelo de Dominio Específico (Mod4CloT) y Lenguaje de Dominio Específico para Arquitecturas CloudIoT

Según Fowler (2010), un lenguaje específico de dominio (DSL) es un lenguaje de programación con expresiones limitadas enfocado en un dominio en particular. Por lo tanto, un DSL es usado por los humanos para dar instrucciones a una computadora sobre lo que debe hacer; además el propósito general es proveer una serie de capacidades: soporte de datos, control y estructuras de abstracción. Un DSL contiene lo mínimo de características necesarias para soportar un dominio; por esta razón al incluirlo en un proceso de desarrollo de una arquitectura de un sistema, permite mejorar la comunicación con los expertos del dominio y por lo tanto acelerar el proceso de desarrollo.

Como resultado, los investigadores y profesionales en los últimos años han desarrollado muchos lenguajes de dominio específico (*Domain Specific Language* - DSL, que se adaptan a dominios de aplicación particulares. Con un DSL apropiado, es posible desarrollar programas de aplicación completos para un dominio de manera más rápida y efectiva que con un lenguaje de propósito general. Idealmente, un DSL bien diseñado captura precisamente la semántica de un dominio de aplicación (Hudak y Cam, 1998). Un DSL aumenta la productividad, reduce los errores, es portátil, tiene un mantenimiento sencillo y permite la reutilización para diferentes propósitos (Deursen, 1997; Deursen, Visser y Klint, 2000). Sin embargo, un DSL tiene una mayor dificultad en su creación, porque los desarrolladores necesitan una abstracción del problema específico (Deursen y Klint, 1998; Deursen et al., 2000; Salihbegovic, Eterovic, Kaljic y Ribic, 2015).

En esta propuesta, hemos creado un modelo de dominio específico, y a partir de éste, un DSL para un editor gráfico; al modelo lo hemos denominado Mod4CloT (*Model for CloudIoT*). Este DSL permitirá especificar aplicaciones para objetos heterogéneos. Con estas aplicaciones, estos objetos podrán conectarse con una plataforma CloudIoT. En este contexto, los usuarios pueden diseñar estas aplicaciones sin necesidad de poseer habilidades de desarrollo y prestar más atención en características del dominio de la

aplicación. El editor gráfico creará objetos y elementos para diferentes plataformas (p. ej. teléfonos inteligentes, computadoras o microcontroladores, infraestructuras en la nube); de este modo se logra abstraer el proceso de desarrollo hasta un metamodelo. Para definir el DSM se ha partido de un metamodelo generado en la herramienta *Eclipse Modeling Framework* (EMF⁴) (Steinberg, Budinsky, Merks y Paternostro, 2008) utilizando una sintaxis abstracta. EMF es útil para la creación de modelos y metamodelos, con facilidades para la generación automática de código y para la construcción de herramientas basadas en modelos. Los metamodelos se serializan en ficheros XML con extensión ecore.

EMF permite generar automáticamente clases Java que implementan elementos de nuestros modelos, además de clases adaptadoras que permiten visualizar y editar un modelo. Las principales características de EMF son:

- Diseñar metamodelos Ecore, con dos tipos de editores:
 - Editor basado en una estructura de tipo árbol.
 - Editor visual, similar a un diagrama de clases UML.
- Construir editores de modelos basados en una estructura de tipo árbol, EMF permite generar clases Java que permiten soportar al metamodelo y clases para pruebas con JUnit.

En la Figura 4-1 se muestra el metamodelo Ecore en el editor visual, también se listan las enumeraciones correspondientes al metamodelo.

Los investigadores y desarrolladores de software, a través de los años, han creado abstracciones que les ayudan a programar en términos de su intención de diseño en lugar del entorno informático subyacente y, por lo tanto, ocultan al mismo tiempo las complejidades de estos entornos. El enfoque principal del desarrollo de software consiste en elevar el nivel de abstracción y el nivel de reutilización (Perisic y Sad, 2014). Un enfoque prometedor para abordar la complejidad de la plataforma y la incapacidad de los lenguajes de tercera generación para aliviar esta complejidad y expresar conceptos de dominio de manera efectiva, es desarrollar tecnologías de soporte impulsadas por modelos (MDE) (Schmidt, 2006). MDE promueve el uso de modelos como artefactos primarios de un proceso de desarrollo de software, en las primeras fases del ciclo de vida del software, como un intento de manejar la complejidad a través de abstracciones. De esta forma, MDE reduce la brecha entre los dominios del problema, solución y ejecución a través del uso de tecnologías que respaldan la transformación sistemática de abstracciones a nivel del problema en implementaciones de software (France, Rumpe, France y Rumpe, 2007). En otras palabras, la aplicación de un proceso MDE da como resultado varios modelos que residen en varios niveles de abstracción.

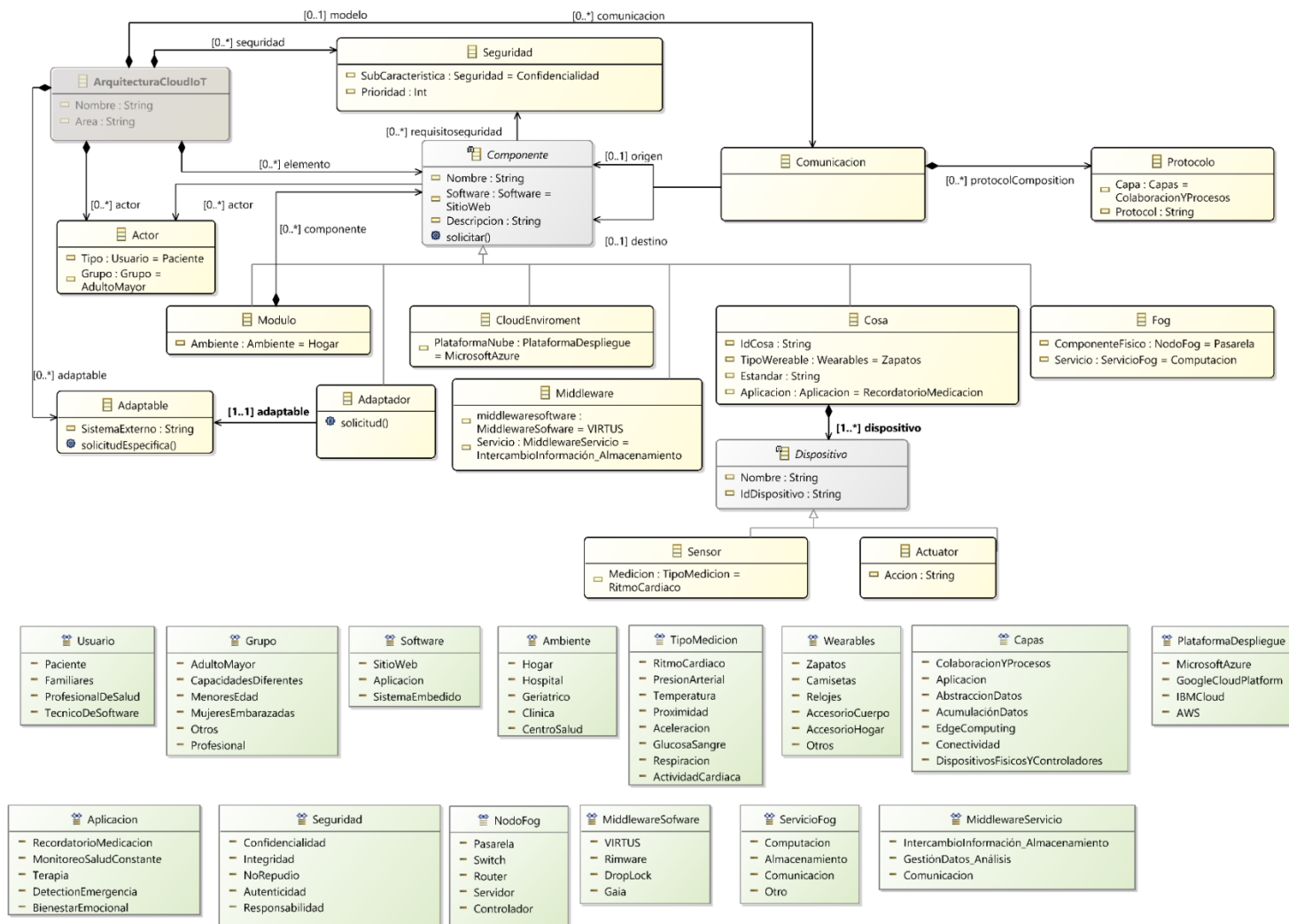


Figura 4-1 DSM Mod4IoT y enumeraciones correspondientes. Fuente: Elaboración propia.

Un problema con los DSL y DSM es que son difíciles de diseñar e implementar y requieren mayores costos iniciales (Liddle, 2011). Aquí es donde entran en contexto las técnicas de MDE: DSL y DSM, las mismas que son una combinación perfecta (Wolterink, 2009). El enfoque MDE hace que la especificación de un DSL sea mucho más fácil. Crear un DSL en el contexto MDE a menudo se llama Modelado Específico de Dominio (DSM) (Wolterink, 2009). A diferencia de los DSL, DSM usa modelos para describir los componentes individuales del sistema de dominio. Los modelos a menudo se basan en una representación gráfica y están respaldados por herramientas de diseño gráfico. Una herramienta que permita generar un DSM (e.j. Eclipse Modeling Framework) permite al usuario crear modelos de dominio y, en general, en función de modelos creados, generar una determinada parte del código. Crear una herramienta gráfica para DSM suele ser un proceso muy complejo. Necesita mucho tiempo para desarrollarse y, a menudo, depende en gran medida de marcos confiables, porque construir un editor desde cero es casi imposible en la actualidad (Vujović, Maksimović y Perišić, 2014).







Una herramienta potente para implementar un DSM y crear un editor gráfico es Sirius (The Eclipse Foundation, 2018). Sirius encapsula el marco de modelado gráfico (GMF) y simplifica el producto, reduce el tiempo de diseño y aumenta rápidamente la productividad general (Vujović et al., 2014).





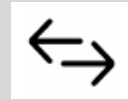
Sirius admite tres dialectos diferentes (tipos de representaciones): (i) diagramas (modeladores gráficos), (ii) tablas y (iii) árboles (representaciones jerárquicas), pero se pueden agregar dialectos nuevos a través de la programación (The Eclipse Foundation, 2018). Debido a que el dominio del problema por lo general hace necesaria la colaboración de personas con diferentes perspectivas, Sirius proporciona la posibilidad de realizar análisis y definir roles y perspectivas de los mismos datos utilizando diferentes puntos de vista en el mismo modelo de dominio.



4.1 Componentes Mod4CloT

En la tabla siguiente, se describe cada uno de los componentes del Mod4CloT presentado en la Figura 4-1, los cuales al interactuar entre sí permiten generar arquitecturas CloudIoT.

Elemento	Descripción del elemento	Representación Gráfica
Arquitectura CloudIoT	Este es el contenedor principal de la arquitectura CloudIoT.	N/D *
Seguridad	Sirve para definir la seguridad de un elemento en el modelo. No es obligatorio y depende del dominio de la arquitectura; tiene una prioridad, en el caso de que un	

	experto en el dominio necesite dar énfasis en una sub característica de seguridad sobre otra. (ISO/IEC, 2011a).	
Componente	Se refiere a cada componente dentro de la arquitectura CloudIoT. Cada componente tiene su software que le permite interactuar con otro componente de la arquitectura.	N/D **
Entorno CloudIoT	Sirve para definir el ambiente de Computación en la Nube de la arquitectura. En esta arquitectura se considera solamente Software como Servicio (SaaS).	
Actor	Son los operadores humanos del sistema.	
Módulo	Un módulo es un conjunto de componentes que tienen una funcionalidad específica. (p. ej. recolección de información, detectar eventos, ejecutar acciones).	
Middleware	El middleware es un habilitador importante que proporciona comunicación entre cosas heterogéneas. Es una capa intermedia entre cosas y servicios de aplicaciones y proporciona una abstracción de la funcionalidad de la cosa para servicios de aplicaciones. (Farahzadi, Shams, Rezazadeh, y Farahbakhsh, 2017)	
Cosa	Es una entidad u objeto físico que tiene un identificador único, un sistema integrado y la capacidad de transferir datos a través de una red. Además, está compuesto por sensores y actuadores.	
Sensor	Es un tipo de dispositivo que debe estar conectado a la red de datos. Un sensor es un objeto que se puede utilizar para medir una propiedad física y convertir esa	

	<p>información en una señal eléctrica u óptica (p. ej. detectar el calor, el peso, el movimiento, la presión, la humedad).</p>	
Actuador	<p>Es un dispositivo capaz de transformar energía en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado (p. ej. dispensar medicamentos, rotar motores).</p>	
Fog	<p>Elemento en el cual el procesamiento de los datos y las aplicaciones se concentran en los dispositivos al borde de la red. De esta forma, los datos pueden ser procesados localmente en un dispositivo inteligente en lugar de ser enviados a la nube. (p. ej. switches, routers, servidores embebidos, pasarela inteligente para procesar y analizar solicitudes y decide si las responde localmente o las envía a la nube). Cualquier dispositivo con computación, almacenamiento y conectividad de red puede ser un nodo Fog (CISCO, 2015).</p>	
Protocolo	<p>Es un sistema de reglas que permite que dos o más elementos se comuniquen. Se trata de las reglas o el estándar que define la sintaxis, semántica y sincronización de la comunicación. Se especifica el protocolo y la capa dentro del modelo de referencia de IoT.</p>	
Comunicación	<p>Se refiere a la conexión de dos elementos dentro de la red para compartir información entre los mismos. Siempre debe existir un elemento origen y destino. Además, se puede añadir un protocolo en la comunicación.</p>	
Adaptador	<p>Este elemento permite que dos interfaces, en principio diferentes, puedan comunicarse. Se encargará de realizar la conversión de una interface a otra. Estas</p>	

Adaptable	interfaces pueden hacer referencia a sistemas o servicios externos.	
	Representa la interfaz externa que se requiere comunicar.	

* Es el contenedor principal de la arquitectura CloudIoT.

** Es una clase abstracta, por lo que no puede ser instanciada.

Tabla 4-1 Componentes del DSM Mod4CIoT. Fuente: Elaboración propia.

Mod4CIoT ha sido propuesto en función de los componentes de la arquitectura, debido a que este modelo permite generar y modificar componentes rápidamente. Además, el mismo podría ser usado para aplicar transformaciones de modelo a modelo y modelo a texto, de esta forma, la utilidad y versatilidad de este modelo permitiría generar soluciones completas, no tan solo a nivel de una arquitectura, sino a todos los niveles en el desarrollo de hardware y de software. Es decir, se podrían generar códigos automáticamente en diferentes lenguajes de programación y en diferentes plataformas de HW y SW.

Capítulo 5

Arquitectura de integración de Computación en la Nube e IoT. Caso de Estudio: Ambient Assisted Living.

Las arquitecturas CloudIoT, se refieren al conjunto de estructuras necesarias para definir un sistema que integre IoT y Computación en la Nube, éstas comprenden componentes de software y las relaciones entre éstos y sus respectivos roles (Bass, Clements, y Kazman, 2003; Cervantes y Kazman, 2016). En el presente capítulo, se ilustra cada una de las fases para el diseño de una arquitectura a partir del Modelo de Dominio Específico descrito en el Capítulo 4, para el efecto, se ha tomado a manera de ilustración el dominio de los ambientes de vida asistidos (AAL).

En base a la investigación realizada en el Capítulo 3, en las arquitecturas CloudIoT se identificaron tres componentes principales comunes:

- 1) Sensores responsables de la recolección de información de la gente y el ambiente (p. ej., datos de salud, ubicación, luminosidad) para estar disponible en la nube.
- 2) Componentes de software ejecutándose en la nube para desempeñar tareas de almacenamiento y procesamiento de información obtenida de los dispositivos IoT.
- 3) Componentes dentro de la capa de red a través de los cuales los datos son transmitidos a los componentes de software en la nube.

La comunicación entre dispositivos y entre estos componentes de software que consumen información se lleva a cabo ya sea, a través de controladores adaptados a cada dispositivo o vía pasarelas inteligentes mediando la comunicación.

Uno de los problemas más importantes y desafiantes es la heterogeneidad de diferentes objetos. Este problema se puede solucionar mediante la implementación de un *middleware* adecuado que se encuentra entre cosas y aplicaciones como una plataforma confiable para la comunicación entre cosas con diferentes interfaces, sistemas operativos y arquitecturas. (Farahzadi et al., 2017).

En el libro *Software Architecture in Practice* (Bass et al., 2003), los atributos de calidad se definen como propiedades medibles o comprobables de un sistema que se utilizan para indicar qué tan bien el sistema satisface las necesidades de sus partes interesadas. Debido a que la calidad tiende a ser un concepto subjetivo en sí mismo, estas propiedades permiten que la calidad se exprese de manera concisa y objetiva. Entre los factores impulsores, los atributos de calidad son los que dan forma más significativa a la arquitectura. La arquitectura de un sistema para ejecutar en la nube, no es diferente a la arquitectura para cualquier otra plataforma distribuida. Es decir, se debe prestar atención a la usabilidad, modificabilidad, interoperabilidad y capacidad de prueba. Los atributos de calidad que tienen algunas diferencias significativas son la seguridad, el rendimiento y la disponibilidad. (Bass et al., 2003)

Los requisitos no funcionales forman parte de la calidad del producto software (p. ej., rendimiento, seguridad, mantenibilidad) los que se encuentran representados en el modelo de calidad, el cual categoriza la calidad del producto en características y subcaracterísticas (ISO/IEC, 2011a, 2011b).

Los sensores y actuadores que conforman las cosas, se utilizan para observar situaciones cambiantes en el entorno. El paradigma CloudIoT dio origen a un nuevo conjunto de servicios y aplicaciones inteligentes, que pueden tener un fuerte impacto en la vida cotidiana. Muchas de las aplicaciones se benefician de las comunicaciones de máquina a máquina (M2M) cuando las cosas necesitan intercambiar información entre ellas y no solo enviarlas a la nube. Los sensores inteligentes o dispositivos capaces de detectar varias dimensiones en un entorno inteligente se combinan para desarrollar un hospital inteligente, un hogar inteligente o cualquier otro entorno para mejorar la vida humana.

Los módulos son un conjunto de componentes que tienen una funcionalidad específica, representan subsistemas de un sistema completo. Debido a la ubicuidad en las aplicaciones CloudIoT es indispensable la integración con sistemas externos. Mediante un adaptador es posible comunicar las interfaces

En base a lo anteriormente descrito, la arquitectura general de sistemas CloudIoT se ilustra en la Figura 5-1. La arquitectura ha sido generada a partir del Mod4CloT. Los componentes Base de Datos, Móvil, Pasarela corresponde a la instancia de elementos Fog, y se encargan de tareas de almacenamiento, comunicación y procesamiento, respectivamente. El componente Cloud corresponde al elemento Entorno CloudIoT. Las flechas indican la dirección del flujo de información y corresponden a la instancia de elementos Comunicación. El Middleware permite comunicar elementos heterogéneos, en este caso Cosas. El componente Módulo está compuesto por Middleware y Cosas, las cuales se componen de actuadores y sensores. Se han instanciado elementos Seguridad (Autenticidad, Confidencialidad), así como Actores (TécnicoDeSoftware).

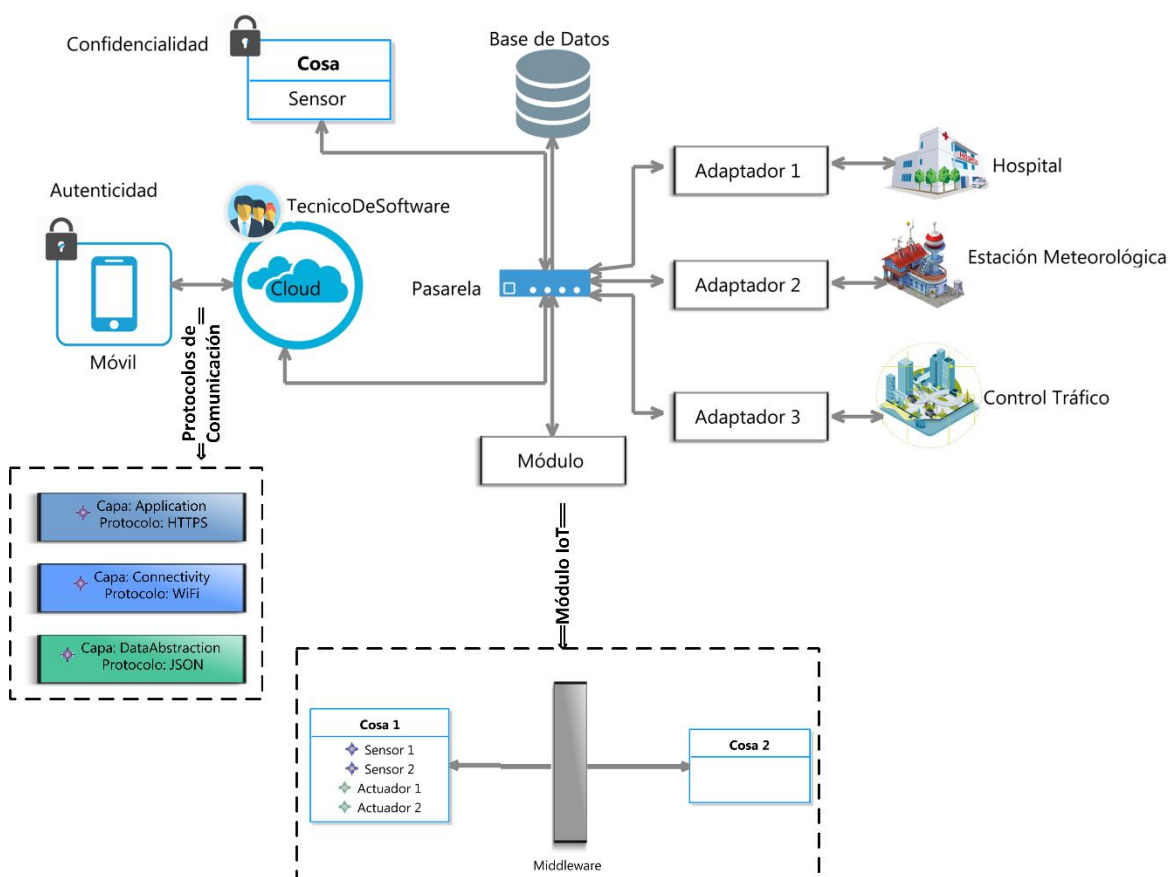


Figura 5-1 Arquitectura general CloudIoT. Fuente: Elaboración propia.

5.1 Caso de estudio: *Ambient Assisted Living (AAL)*

En esta sección se presenta un ejemplo del uso de Mod4CloT. Primero presentamos el contexto del sistema y luego examinamos el diseño arquitectónico. Varios métodos de diseño de arquitectura han sido propuestos y documentados, sin embargo se ha elegido el método Diseño basado en atributos (Attribute-Driven Design, ADD) (Cervantes y Kazman, 2016) debido a que es más concreto y específico en sus pasos y guía para llevar a cabo la actividad de diseño de arquitectura.

5.1.1 Descripción y contextualización del caso de estudio

Este caso de estudio involucra el área de la salud, concretamente un ambiente de vida asistido. La elección de este dominio se realizó debido a la fuerte integración tecnológica que suponen estos escenarios en los cuales, la entrega adecuada de los servicios de salud junto con la supervisión del paciente se consideran cuestiones clave para mejorar la calidad de vida y garantizar una atención de salud y social eficiente (Doukas et al., 2012). Una aplicación IoT en el dominio de salud consta de componentes heterogéneos y se utiliza para controlar la temperatura, humedad, presión, oxígeno, dióxido de carbono, frecuencia cardíaca, frecuencia respiratoria, entre otros. Además, un sistema heterogéneo, como un sistema AAL, puede integrar diversos sensores o dispositivos producidos con diferentes especificaciones. Por lo tanto, es necesario diseñar una arquitectura común para abordar los problemas de integración y comunicación (Cubo, Nieto, y Pimentel, 2014).

Consideramos un caso concreto de monitoreo remoto en el hogar, en el cual se monitorean periódicamente varios signos vitales de Alicia, una paciente que se enmarca dentro de la población de adultos mayores con problemas cardíacos. Además, debido a los problemas cardíacos de Alicia, se debe implementar un mecanismo de informe acerca de cualquier caída accidental mientras se encuentra sola en la habitación del hospital, así como también la necesidad de monitorear el cumplimiento del consumo adecuado de su medicación en su casa.

El objetivo es monitorear a Alicia y, a través del análisis de la información detectada, detectar situaciones que pueden interpretarse como una caída o valores anormales en los signos vitales. Por lo tanto, si se detecta una caída, las acciones se desencadenan automáticamente, como la emisión de una alerta al centro de atención o el control de algunos de sus signos vitales (por medio de sensores siempre conectados a ella). Información útil necesaria para detectar una caída es la ubicación de Alicia y la posición detectada mediante un acelerómetro y un giroscopio. Por lo tanto, el entorno debe estar equipado con unidades de detección que capturen esta información junto con actuadores



capaces de contactar con el centro de atención remota siempre que se detecte una caída o un movimiento extraño, de este modo, en el centro de atención se analiza la información recibida en tiempo real y se contactan a los profesionales de la salud en caso de detectar una situación de emergencia o de movimiento inusual.

Dentro de nuestro escenario, para el control de la medicación su hogar debe contar con un pastillero digital programable desde el celular (Parra, Valdez, Guevara, Cedillo, y Ortiz-Segarra, 2017) de sus cuidadores, almacenando información para el control que debe ser analizada por el médico o profesional de salud a su cuidado.

En el escenario elegido, el sistema está compuesto por sensores y dispositivos conectados al paciente, para detectar y prevenir cualquier situación de emergencia, tanto en el hospital como en el hogar. Por lo tanto, los usuarios finales del sistema son:

Los pacientes, que pueden estar situados en el hospital o en el hogar. Las diferentes salas del hospital y también el hogar del paciente monitoreado pueden tener un conjunto de dispositivos disponibles como servicios conectados al sistema de monitoreo de emergencia (a través de una red local o Internet). Los profesionales de la salud o especialistas (generalmente un médico o una enfermera), se encuentran en sus oficinas o en las salas de emergencia. Cada especialista puede acceder a un conjunto de dispositivos disponibles como servicios que envían información relacionada con los pacientes. Los cuidadores supervisan al paciente en el hogar, corresponden a los familiares cercanos y enfermeras personales. Los centros de cuidado remota, que se encargan de ayudar de forma remota a los profesionales de la salud a ayudar a los pacientes que se monitorean, con un procesamiento intermedio de la información. Los centros de atención pueden ser administrados desde el hospital o externamente, y no siempre necesitan ser parte del sistema.

El sistema está formado, por lo tanto, los siguientes componentes principales:

Sensores para medir los signos vitales básicos de los pacientes, como las señales cardíacas.

Dispositivos o cosas con un comportamiento más complejo, utilizado para detectar o para controlar algunas otras acciones del paciente (análisis de signos vitales). Pueden comprender varios sensores integrados e interactuando entre sí (por ejemplo, el distintivo de usuario o habitación de un paciente, donde varios sensores están conectados, trabajando juntos).

Una *aplicación de monitoreo*, utilizada por los profesionales médicos o los centros de atención, para acceder a todos los sensores y dispositivos

conectados, así como para administrar la información monitoreada; que puede desarrollarse como una aplicación móvil o de escritorio para monitoreo local (en el hospital) o como una aplicación en la nube con la capacidad de monitoreo remoto (en el hogar o también en el hospital conectando a la nube).

Un *repositorio de datos* alojado en una plataforma en la nube para registrar a los usuarios y almacenar y gestionar la información detectada y monitoreada del sistema en cualquier momento dado; que contiene el registro del perfil del usuario (pacientes, especialistas y también las personas que trabajan en los centros de atención remota), sensores y servicios (inactivo y activo, es decir, conectado a un paciente que se está supervisando), se utiliza como el punto de procesamiento central para las solicitudes y desencadenantes históricos de todos los datos y eventos, y también es responsable de mantener el cronograma del paciente.

5.2 Requisitos del sistema

Las actividades de obtención de requisitos se han realizado previamente. Los requisitos más importantes han sido identificados. Comprenden un conjunto de casos de uso primarios, un conjunto de escenarios de atributos de calidad, un conjunto de restricciones y un conjunto de asuntos arquitectónicos. Este conjunto de información es la entrada del paso 1 en el proceso de diseño de la arquitectura, ver Tabla 5-1.

Las restricciones pueden tomar la forma de tecnologías obligatorias, otros sistemas con los que el sistema necesita interoperar o integrar, leyes y estándares que deben cumplirse, las capacidades y la disponibilidad de los desarrolladores, fechas límites que no son negociables, compatibilidad con versiones anteriores versiones de sistemas, y así sucesivamente. Los asuntos arquitectónicos (*Architectural Concerns*) abarcan aspectos adicionales que deben considerarse como parte del diseño arquitectónico, pero que no se expresan como requisitos tradicionales (Cervantes y Kazman, 2016).

5.3 Proceso de diseño de la arquitectura

En esta sección se listan las fases seguidas en el proceso de diseño de la arquitectura del sistema, siguiendo el método *AAD* (Cervantes y Kazman, 2016). Se ha hecho uso de *SPEM 2.0 (Software & Systems Process Engineering Meta-Model Specification)* (OMG, 2008) para representar gráficamente las fases del método *AAD*, se ha incluido el uso de *Mod4CIoT* para instanciar los elementos de la arquitectura (ver Figura 5-2).

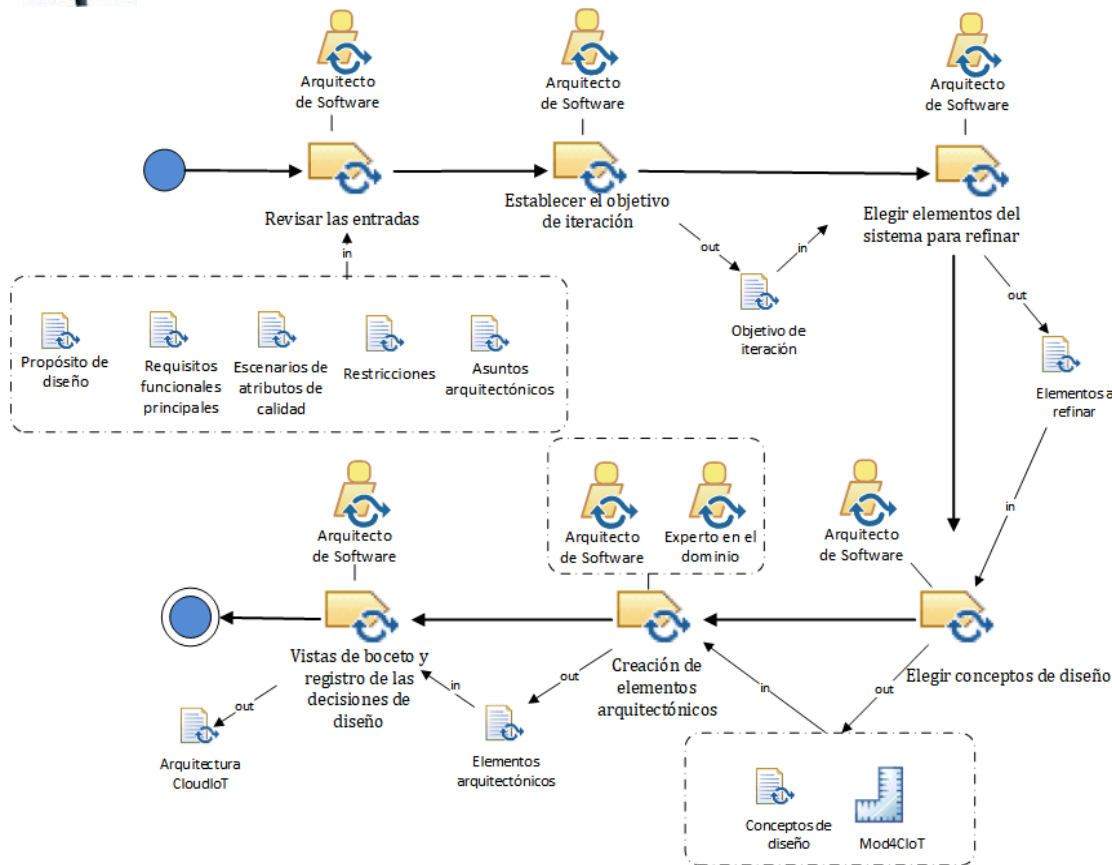


Figura 5-2 Proceso de diseño de la una arquitectura CloudIoT. Fuente: Elaboración propia.

5.3.1 Revisar las entradas

Antes de comenzar el diseño, es necesario asegurarse de que las entradas al proceso de diseño están disponibles y son correctas, se analizan también los factores determinantes de la arquitectura, que "impulsan" las decisiones de diseño, algunas de las cuales están documentadas como requisitos, pero muchas de las cuales no lo son. Las entradas de este paso se resumen en la siguiente tabla.

Categoría	Detalles
Propósito de diseño	Este es un sistema CloudIoT del dominio de salud. El propósito es producir un diseño de la arquitectura del sistema de monitoreo planteado en un escenario de AAL.
Requisitos funcionales principales	<p>Del caso presentado en la sección 5.1, los siguientes son los requisitos principales:</p> <ul style="list-style-type: none"> • Cuando se detecta un movimiento rápido a través de los acelerómetros o giroscopio del dispositivo de usuario, inmediatamente el sistema debe comprobar la frecuencia cardíaca

	<p>(debe controlarse periódicamente).</p> <ul style="list-style-type: none">• La información de los sensores debe ser enviada a un dispositivo de procesamiento para ser analizada y por el cual se puede enviar esta información a la nube y al centro de cuidado.• En el caso de que determine (mediante el análisis correspondiente) que puede haber ocurrido una caída del paciente, se realiza una notificación automática al centro de atención.• El centro de cuidados debe analizar la información recibida mediante los dispositivos del hospital (historial y en tiempo real), y contactar a los profesionales de la salud en caso de detectar una situación de emergencia.• La información de los dispositivos o cosas de los hogares y hospital es enviada a la nube, donde se almacena y gestionar la información detectada y monitoreada.• La aplicación de monitoreo debe presentar al profesional de salud información del paciente: cronograma del consumo y gestión medicamentos, historial de pacientes, información de los dispositivos y sensores correspondiente a cada paciente.• El pastillero digital deber permitir la gestión de medicamentos (hora, dosis, medicamento) programada mediante una aplicación móvil.
Escenarios de atributos de calidad	<p>Los atributos de calidad principales en el escenario son la seguridad y la disponibilidad, debido a su nivel de importancia en arquitecturas de Computación en la Nube (Cervantes y Kazman, 2016), así como la protección de datos y privacidad en aplicaciones de salud (Kuo, 2011). Estos atributos han sido incluidos como factores.</p>
Restricciones	<p>A continuación, se muestran las restricciones identificadas para diseñar la arquitectura, incluidas como factores.</p> <ul style="list-style-type: none">• El sistema deberá ser compuesto principalmente de tecnologías <i>open source</i> (por motivos de costo). Para aquellos componentes en donde la relación valor/costo del uso de tecnologías propietarias es alto, tecnologías

Asuntos arquitectónicos	<p>propietarias pueden ser utilizadas.</p> <ul style="list-style-type: none">• Se debe controlar el nivel de acceso de los usuarios a través de la asignación de privilegios.• La aplicación de monitoreo debe ser accedida mediante un explorador web.• Una política de respaldo de información debe ser manejada para el repositorio de datos.• Los datos de los sensores del paciente deben ser analizados en tiempo real.• El sistema debe conectarse con el centro de cuidado, el cual es un sistema externo.
	<p>A continuación, se listan los asuntos arquitectónicos incluidas como factores.</p> <ul style="list-style-type: none">• Colocar las funcionalidades específicas en módulos.• Aprovechar el conocimiento del equipo en aplicaciones CloudIoT.

Tabla 5-1 Entradas del proceso de diseño. Fuente: Elaboración propia.

5.3.2 Establecer el objetivo de iteración

Esta es la primera iteración en el diseño del sistema, el objetivo de la iteración es establecer una estructura para el sistema cuyos requisitos han sido expuestos anteriormente. Se debe tener en cuenta todos los factores y, en particular, las limitaciones y los atributos de calidad.

5.3.3 Elegir uno o más elementos del sistema para refinar

Debido a que se trata de un desarrollo totalmente nuevo, y es la iteración inicial, el elemento a refinar es todo el sistema. Como se presentó anteriormente los elementos principales del sistema son: sensores y dispositivos, plataforma de Computación en la Nube, aplicación de monitoreo.

5.3.4 Elegir uno o más conceptos de diseño que satisfagan los factores seleccionados

La siguiente tabla resume las decisiones de diseño tomadas con respecto a la selección de conceptos de diseño. En el libro *Application Architecture Guide* (Microsoft Patterns y Practices Team, 2009) se presentan patrones de diseño relevantes en aplicaciones de Computación en la Nube, también se ha considerado el modelo de referencia para aplicaciones IoT (Green, 2014).

Decisiones de diseño y ubicación	Razón fundamental
Descomposición en módulos Aplicación Web.	Para ayudar a lograr alta modificabilidad, lo que significa alta cohesión y bajo acoplamiento. Puede ser accedido desde cualquier dispositivo inteligente, utilizando un explorador web
Usar una nube privada	Para ayudar a lograr una alta disponibilidad, en caso de que en el hogar se pierda la conexión con internet. Logrando así la tolerancia a fallos y disponibilidad.
Proveedor de servicios de Computación en la Nube.	La plataforma en la nube permite manejar la información de monitoreo de una manera rápida, simple y segura. La seguridad de la plataforma radica en la seguridad del proveedor del servicio.
Integración del patrón de diseño adaptador	Este elemento permite que el sistema se conecte con el Centro de Cuidados, el cual es un sistema externo.
Comunicación dúplex	Comunicación de mensajes bidireccionales donde tanto el servicio como el cliente se envían mensajes entre sí de forma independiente, importante en un sistema que integre IoT y la Computación en la Nube.
Control de objetos físicos o cosas	Se ocupa de la identificación y recolección de información específica de los objetos por medio dispositivos sensores.
Servicio Fog	Estos servicios dirigen los datos al lugar correcto para el análisis: en la Nube o en el componente Fog.
Seguridad de datos	Para ayudar a lograr la seguridad, se limitan los puntos de acceso a recursos críticos y se aplican protocolos seguros de transmisión de información.

Tabla 5-2 Conceptos de diseño seleccionados. Fuente: Elaboración propia.

5.3.5 Creación de instancias de elementos arquitectónicos, asignación de responsabilidades y definición de interfaces

Las decisiones de diseño instanciadas consideradas en la fase anterior, se resumen en la tabla 5-3. Cada elemento hace referencia a los componentes CloudIoT que forman parte del Mod4ClouIoT, los cuales han sido seleccionados de acuerdo con las decisiones de diseño realizadas. Un experto en el dominio participa en este proceso, se encarga de elegir e instanciar los componentes de adecuados de Mod4ClouIoT para generar la arquitectura.

Decisiones de diseño y ubicación	Razón fundamental
Cosas: Sensores y	Cada cosa en IoT está compuesta por sensores y

actuadores.	actuadores, cada una estos elementos (pastillero digital, monitor de caídas, monitor cardiaco) tiene una funcionalidad específica.
Módulos: Hogar y Hospital.	Ya que el sistema interactúa en dos ambientes diferentes deben existir módulos que agrupen funcionalidades correspondientes.
Comunicación: Protocolo MQTT	Comunicación entre cosas y elementos Fog (móvil, pasarela), así como estos elementos Fog y la Nube. MQTT es un protocolo de mensajería ligero para pequeños sensores y dispositivos móviles.
Comunicación: Protocolo HTTP	Uso de REST para comunicarse con el sistema externo y la aplicación de monitoreo. HTTP es utilizado para comunicarse con recursos de Internet o cualquier aplicación con un cliente de navegador web.
Comunicación: TLS	Agrega seguridad a la capa de transporte para la protección de datos y transmisión segura. Además, trabaja junto a HTTP y MQTT.
Fog: Pasarela, móvil, arduino, waspmote.	Tienen el objetivo de monitorear o analizar datos en tiempo real de cosas conectadas a la red y luego iniciar una acción. La acción puede involucrar comunicaciones de máquina a máquina (M2M) o interacción hombre-máquina (HMI).
Computación en la Nube: Microsoft azure	Soporta la integración de MQTT y HTTP. Implementa un estilo arquitectónico basado en eventos, aplicados especialmente en el dominio de IoT y sistemas en tiempo real (Microsoft, 2018).
Actores: paciente y profesional de la salud.	Identificar a los pacientes y profesionales de la salud para permitir el acceso por privilegios y aumentar la seguridad.
Adaptable: Centro de cuidado	El centro de cuidados es tratado como un sistema externo, y es integrado al sistema mediante un adaptador. Las interfaces del centro de cuidado y el hospital se comunican por medio de este componente.

Tabla 5-3 Casos de uso del sistema. Fuente: Elaboración propia.

5.3.6 Vistas de boceto y registro de las decisiones de diseño

En la figura 5-3 se presenta el gráfico de la arquitectura propuesta para el caso de estudio AAL.

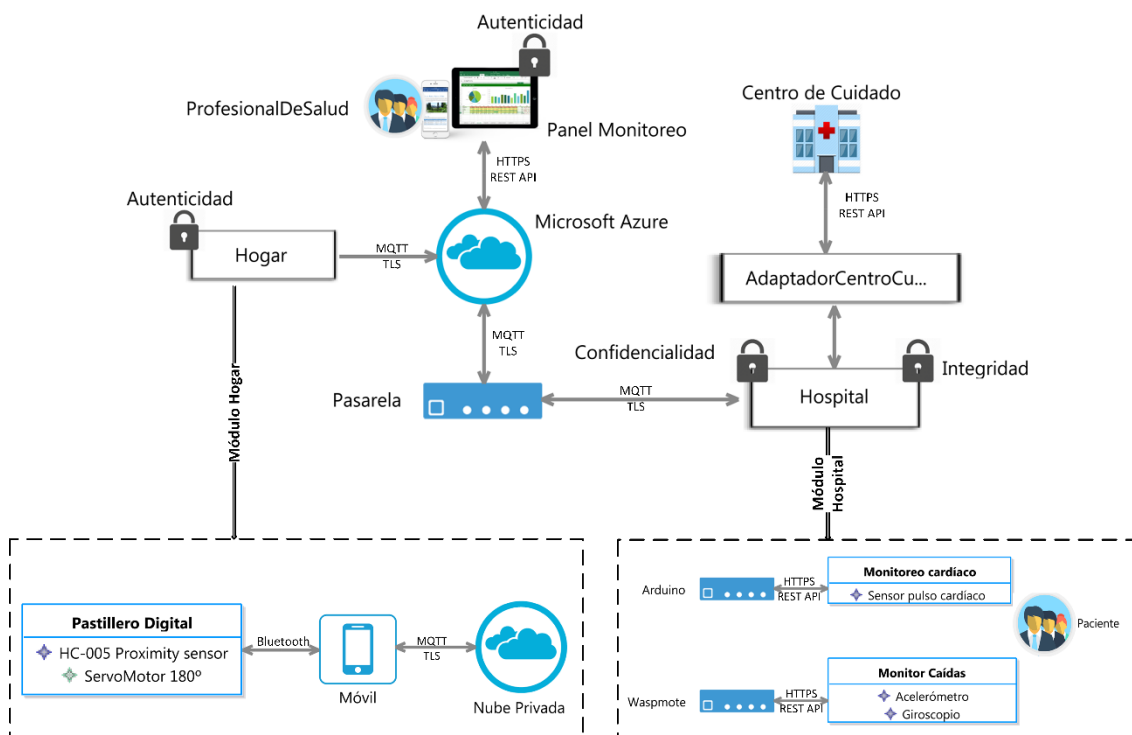


Figura 5-3 Arquitectura obtenida del caso de estudio. Fuente: Elaboración propia.

Las responsabilidades de los elementos representados en el gráfico anterior se describen en la siguiente tabla.

Elemento	Responsabilidad
Centro de cuidado	Sistema externo a través del cual un profesional de la salud recibe información del paciente en tiempo real cuando sucede un accidente.
Adaptador centro de cuidado	Componente que permite que las interfaces del centro de cuidado y del módulo hospital se comuniquen, debido que son sistemas distintos.
Panel de monitoreo	Este componente utiliza la API externa publicada por la plataforma en la nube para desarrollar aplicaciones como una interfaz web para la gestión de datos personales o la supervisión remota de pacientes e incluso clientes móviles para realizar cualquier operación disponible en la API (p. ej., recuperar la información de Alicia).
Nube Microsoft Azure	La plataforma en la nube (a nivel de datos): recolecta información presionada de la comunicación de red utilizando una API REST, y su función principal es facilitar la administración remota (almacenamiento,

	acceso y monitoreo) de los sensores datos en tiempo de ejecución, lo que permite el acceso simplificado del usuario y el manejo efectivo de la elasticidad de la demanda, reaccionando a ciertas situaciones de acuerdo con los resultados del monitoreo.
Pasarela	Se encarga de comunicar todos los dispositivos heterogéneos (a través de TCP / IP). Conecta el subsistema del hospital con la plataforma en la nube.
Módulo Hogar	En caso de que el módulo hogar pierda conexión a Internet, sus componentes funcionan normalmente debido a que la nube privada gestiona la toma de medicamentos.
Pastillero digital Móvil	Gestionar la medicación de los pacientes. Dispositivo Fog por el cual se programa el pastillero digital (alarmas y tomas de medicación de los pacientes).
Nube Privada	Plataforma que almacena el historial de medicamentos de los pacientes, para presentar a los profesionales de la salud.
Módulo Hospital	Subsistema que se encarga de monitorizar y analizar la información de paciente en el hospital.
Arduino	Dispositivo que se encarga de analizar la información recibida por el monitor de caídas, envía información del estado del paciente al centro de cuidados. Se comunica con el repositorio digital de datos en la nube.
Waspnote	Plataforma de sensores que se encarga de analizar la información recibida por el monitor cardiaco, envía información del estado del paciente al centro de cuidados. Se comunica con el repositorio digital de datos en la nube.
Monitor Cardiaco	<i>Wearable</i> usado por el paciente para registrar la actividad eléctrica de su corazón
Monitor Caídas	<i>Wearable</i> usado por el paciente, registra los movimientos realizados para detectar caídas.
Actor: Paciente y profesional de la salud.	Actores del sistema que cuentan con una identificación y contraseñas únicas.

Tabla 5-4 Casos de uso del sistema. Fuente: Elaboración propia.

Como todos los controladores fueron completamente tratados, solo se requirió una iteración del método ADD.



La arquitectura propuesta para el caso de estudio revisado en el presente capítulo ha sido evaluada mediante un cuasi-experimento a ser presentado en el Capítulo 6, siguiendo los lineamientos de Wohlin et al. (2012).

Capítulo 6

Evaluación de la arquitectura propuesta

Este capítulo describe un cuasi-experimento que tiene como objetivo evaluar la arquitectura CloudIoT, teniendo en cuenta la percepción de usuario al momento de realizar cambios sobre la misma. En la sección 6.1 se presenta una introducción a la evaluación empírica de la arquitectura propuesta. En la sección 6.2 se muestran los modelos teóricos de evaluación en Ingeniería del Software. La sección 6.3 presenta la adaptación del modelo de evaluación a ser utilizado en la arquitectura. La sección 6.4 muestra un cuasi-experimento de la arquitectura propuesta en esta tesis. La sección 6.5 presenta las amenazas de validez y finalmente la sección 6.6 presenta las conclusiones de este capítulo.

6.1 Introducción

Teniendo en cuenta el alto impacto que las percepciones de los usuarios tienen en la selección y adopción de una nueva solución, la aplicación del Method Evaluation Model (MEM) (Moody, 2001) puede ser de utilidad a la hora de evaluar las percepciones de un usuario frente a la adopción de una nueva tecnología, además de la eficacia y efectividad de los usuarios frente a la misma, en este caso la Arquitectura CloudIoT propuesta. El MEM fue originalmente propuesto como un medio para evaluar métodos de diseño de Sistemas de Información; sin embargo, éste ha sido exitosamente aplicado en la evaluación de otro tipo de métodos (Abrahão, Insfran, Carsí y Genero, 2011; Cedillo, 2016; Cedillo Orellana, Insfrán Pelozo y Abrahao, 2017) Este modelo ha sido elegido debido a que integra variables del rendimiento actual de los usuarios y de su percepción como mecanismo para predecir la intención de uso y posible adopción en práctica de un método. MEM extiende el Technology Acceptance Model (TAM) (Davis, 1985), el cual ha sido validado empíricamente en numerosos estudios que demuestran su utilidad para analizar la facilidad de uso percibida, utilidad percibida e intención de uso de los participantes aplicando un método para predecir su posible aceptación.

Para utilizar el MEM en la evaluación de una arquitectura CloudIoT es necesario operacionalizar este modelo teórico. Esta operacionalización consiste en definir los constructores del modelo en función de las variables

relevantes para evaluar arquitecturas y redefinir los ítems del cuestionario que permitirán medir las variables de percepción.

De esta forma, este capítulo presenta la operacionalización realizada, así como el diseño y ejecución del cuasi-experimento realizado para proporcionar evidencias sobre la utilidad de una arquitectura CloudIoT. Los usuarios utilizaron una instancia de la arquitectura, la cual fue presentada en el Capítulo 5 para visualizar su diseño y realizar cambios sobre la misma.

6.2 Modelos teóricos de evaluación en Ingeniería de Software

El Technology Acceptance Model (TAM) propuesto por Davis (Davis, 1985) es un modelo teórico ampliamente usado desde la perspectiva del uso en general de un sistema.

El Method Evaluation Model (MEM) propuesto por Moody (Moody, 2001) provee mecanismos para evaluar el rendimiento actual, la aceptación y la posible adopción de un método de sistemas de información en la práctica.

A continuación, se presentan los modelos teóricos más relevantes para el propósito de este trabajo de titulación.

6.2.1 Technology acceptance model (TAM)

El TAM constituye una adaptación de la teoría Theory of Reasoned Action (TRA) propuesta por Fishbein y Ajzen (1975).

El TAM utiliza dos creencias de adaptadores potenciales, la facilidad de uso percibida y la utilidad percibida de la tecnología como los principales determinantes de las actitudes hacia una nueva tecnología. Estas actitudes influyen en las intenciones y de ahí el comportamiento. En este modelo, el uso es modelado como una función directa de la intención (Ver la Figura 6-1).

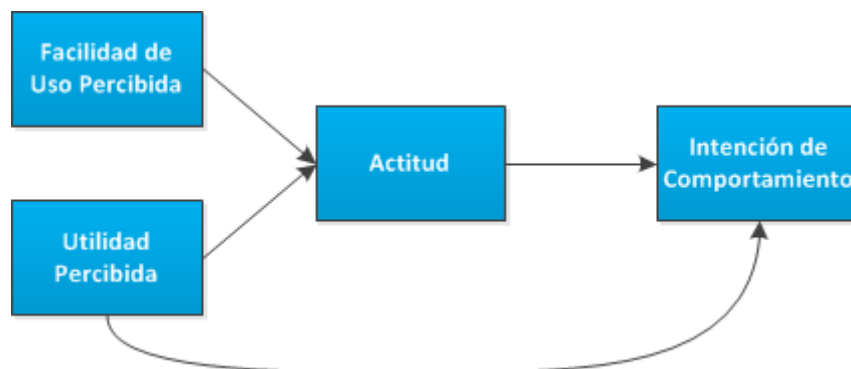


Figura 6-1 Technology Acceptance Model (TAM) simplificado. Fuente: Riemenschneider, Hardgrave, & Davis (2002).

El significado de cada constructor de TAM es:

- Utilidad Percibida (PU): la probabilidad subjetiva del usuario de que utilizando una aplicación específica podría incrementar su rendimiento laboral en un contexto organizacional.
- Actitud (A): el deseo del usuario para usar el sistema. Tanto PU como PEOU predicen la actitud hacia usar el sistema.
- Intención de Comportamiento (IC): la medida de la resistencia a ejecutar un comportamiento específico. A y PU influyen al individuo de IC a usar el sistema.
- Uso: el uso actual del sistema. Este es predicho por intenciones del comportamiento.

6.2.2 Method Evaluation Model (MEM)

El MEM incorpora dos aspectos del método de éxito: la eficiencia actual y el uso actual. La Figura 6-2 ilustra las diferentes partes de este método. Ambos aspectos deben ser considerados para evaluar arquitecturas CloudIoT. La figura además muestra los constructores del modelo, así como también las relaciones causales a lo largo de los constructores del mismo.

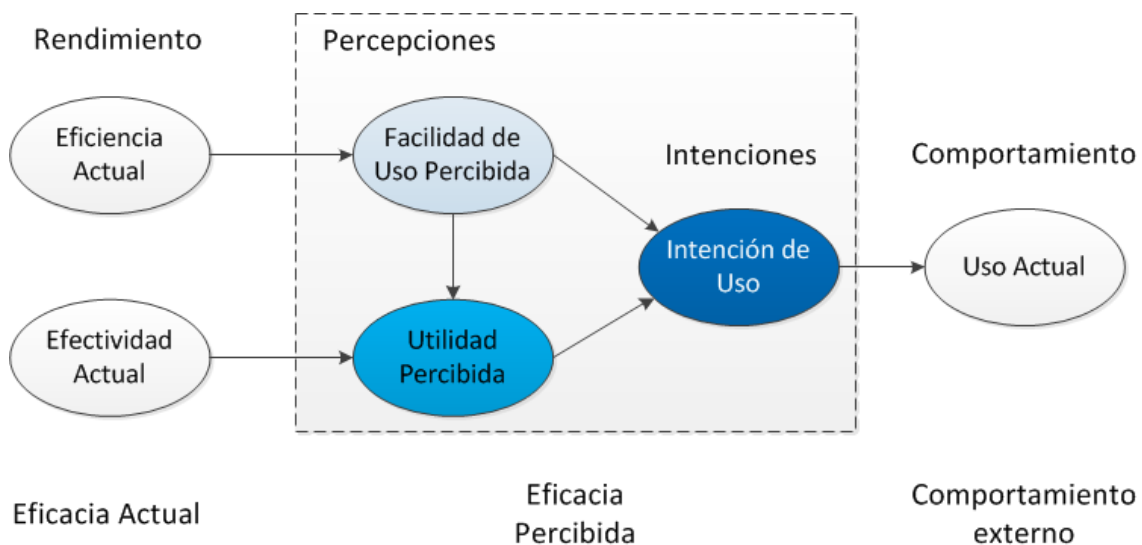


Figura 6-2 Method Evaluation Model – MEM. Fuente: Moody (2001)

Los constructores del MEM están basados en el Technology Acceptance Model (TAM), un método bien conocido y validado empíricamente para la evaluación de tecnologías de la información. Los constructores del MEM son:

- Eficacia Actual: este constructor tiene dos variables basadas en el rendimiento del usuario:
 - Eficiencia Actual: es el esfuerzo requerido para aplicar un método.
 - Efectividad Actual: es el grado en el cual un método consigue sus objetivos. Este constructor está relacionado a la calidad de los artefactos obtenidos al momento de aplicar el método. Según

Rescher (Gomberg y Rescher, 1975), todos los métodos intentan conseguir ciertos objetivos. Rescher define un método como “una colección de reglas y procedimientos designados para asistir a la gente al momento de ejecutar una acción particular”

- Eficacia Percibida, la cual tiene dos variables basadas en la percepción:
 - Facilidad de Uso Percibida: es el grado en el cual una persona cree que usando un método en particular puede estar libre de esfuerzo.
 - Utilidad Percibida: es el grado en el cual una persona cree que usando un método particular podría mejorar su rendimiento en el trabajo.
- Intención de Uso: es el modo en el que una persona intenta usar un método particular. Esta variable representa un juicio perceptual de la eficacia y efectividad de coste del método. Esta variable es usada para medir la probabilidad del método para ser adoptado en la práctica. Las relaciones causales sugeridas que perciben la facilidad de uso y la utilidad percibida directamente afectan la intención de usar el método.
- Uso Actual: el cual representa una variable basada en el comportamiento, definida como la intención de utilizar un método en la práctica. De acuerdo a la relación causal hipotética, el uso actual debe estar determinado por la intención del uso.

6.3 Adaptando el MEM para evaluar Arquitecturas CloudIoT

- El primer paso para adaptar el MEM es definir los objetivos específicos de las arquitecturas CloudIoT. Los constructores generales del MEM pueden ser instanciados en variables dependientes concretas basadas en estos objetivos.
- De las arquitecturas CloudIoT existentes se aprecian dos objetivos principales: (1) evaluar la instancia de la arquitectura según un dominio específico y realizar modificaciones sobre ella en base a necesidades particulares y (2) generar soluciones que mejoren la eficacia en la implementación de dichas soluciones. Este capítulo se centra en el primer objetivo, ya que esta es una actividad común en arquitecturas CloudIoT. Las siguientes variables de rendimiento son utilizadas para medir la eficiencia y la efectividad de los usuarios con respecto a el análisis y modificaciones en la arquitectura CloudIoT.
- Efectividad actual: la proporción entre el número de modificaciones correctamente realizadas en la arquitectura CloudIoT y el número total de modificaciones a ser realizadas.

$$\text{Efectividad} = \frac{\text{\#Modificaciones correctamente realizadas}}{\text{Número total de modificaciones a ser realizadas}}$$

- Eficiencia actual: el tiempo empleado para realizar las modificaciones.

$$Eficiencia = \sum_{i=1}^n \text{Tiempo utilizado en realizar la modificación}_i$$

Con el objetivo de medir las variables basadas en la percepción, se ha adaptado un instrumento de medición utilizado en el MEM. La figura 6-2 muestra cómo el MEM ha sido operacionalizado para evaluar arquitecturas CloudIoT. Para medir cada uno de los tres constructores, se han definido conjuntos de preguntas basados en los ítems mostrados en la Tabla 6-1. La figura 6-3 muestra el modelo teórico propuesto para evaluar arquitecturas CloudIoT. Básicamente, se han usado medidas basadas en el rendimiento como factores que influencia las variables basadas en la percepción (ver Figura 6.4).

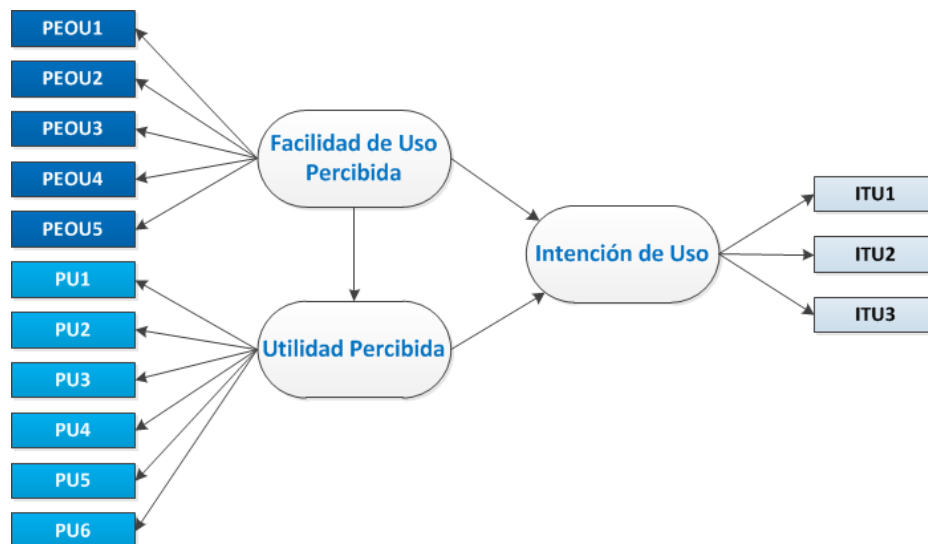


Figura 6-3 Distribución de preguntas del cuestionario aplicado al cuasi-experimento. Fuente: Elaboración propia.

De acuerdo a nuestra adaptación de MEM, la probabilidad de que una arquitectura CloudIoT pueda ser aceptada en la práctica puede ser predicho probando las siguientes hipótesis.

- $H1_0$: La arquitectura CloudIoT se percibe como difícil de usar, $H1_0 = \neg H1_1$
- $H2_0$: La arquitectura CloudIoT no se percibe como útil, $H2_0 = \neg H2_1$
- $H3_0$: No existe intención de utilizar esta arquitectura CloudIoT en el futuro, $H3_0 = \neg H3_1$

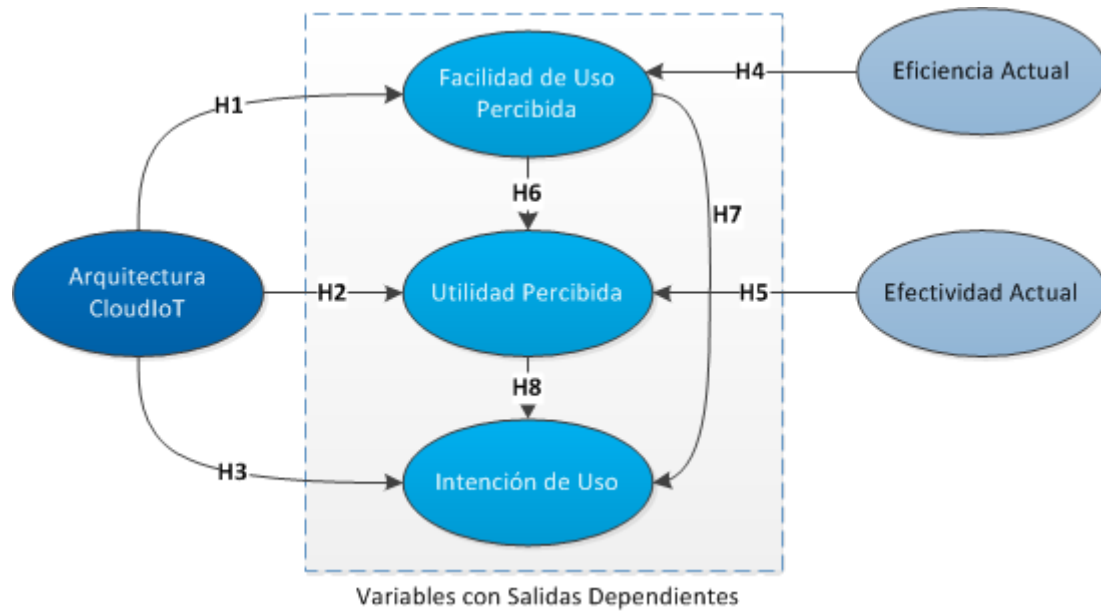


Figura 6-4 Modelo teórico para la evaluación de arquitecturas CloudIoT. Fuente: Elaboración propia.

A continuación, se presentan hipótesis que muestran una relación directa entre el uso de una arquitectura en particular y el rendimiento, percepciones e intenciones de los usuarios. El modelo de evaluación además propone un número de hipótesis que indican una relación causal entre las variables dependientes.

- $H4_0$: La facilidad de uso percibida no puede verse determinada por la eficiencia, $H4_0 = \neg H4_1$. La razón de esta hipótesis es que la eficiencia representa una medida basada en el rendimiento de la eficiencia actual, mientras que la facilidad de uso percibida representa una medida basada en la percepción. De acuerdo a MEM, la eficiencia mide el esfuerzo requerido para aplicar el método, el cual debería determinar las percepciones del esfuerzo requerido.
- $H5_0$: La percepción de la utilidad no está determinada por la efectividad, $H5_0 = \neg H5_1$. La razón de esta hipótesis es que la efectividad representa una medida basada en el rendimiento, mientras que la utilidad percibida representa una medida basada en la percepción de la efectividad. De acuerdo al MEM las percepciones de efectividad deberían estar determinadas por la efectividad actual.

- $H6_0$: La utilidad percibida no es determinada por la facilidad de uso, $H6_0 = \neg H6_1$. La razón de esta hipótesis es que la facilidad de uso percibida representa una medida basada en la percepción, mientras que la utilidad percibida representa una medida basada en el rendimiento. De acuerdo al MEM las percepciones de facilidad de uso deberían estar determinadas por la facilidad de uso actual.

percibida, $H6_0 = \neg H6_1$. Esta hipótesis es tomada desde el TAM, en el cual la facilidad de uso percibida se encuentra que no tiene una influencia directa sobre la utilidad percibida.

- *H7₀: La intención de uso no es determinada por la facilidad de uso percibida*, $H7_0 = \neg H7_1$. Esta hipótesis es tomada desde el TAM en el cual se encuentra que la facilidad de uso percibida tiene influencia sobre la intención de uso.
- *H8₀: La intención de uso no está determinada por la utilidad percibida*, $H8_0 = \neg H8_1$. Esta hipótesis es tomada del TAM, en la cual se encontró que la utilidad percibida tiene una influencia directa sobre la intención de uso.

La Tabla 6-1 muestra los ítems definidos para medir las variables basadas en la percepción. Estos ítems fueron combinados en un cuestionario con 14 preguntas. Los ítems fueron formulados utilizando una escala de 5 puntos de Likert, con el formato de preguntas opuestas. Varios ítems con el mismo grupo de constructores fueron aleatorizados para prevenir errores de respuesta sistemática. El instrumento de medición está disponible en <http://bit.ly/2NDKkQ9>.

Pregunta	Declaración Positiva (5 puntos)
PEOU1	La forma para crear y modificar la arquitectura CloudIoT me ha parecido compleja y difícil de seguir
PEOU2	De manera general, la arquitectura CloudIoT (sus componentes y elementos) es difícil de entender
PEOU3	La manera de modificar los requisitos durante la construcción de una solución utilizando la arquitectura CloudIoT son claros y fáciles de entender
PEOU4	La forma en la que se realizan modificaciones en las instancias de la arquitectura CloudIoT es difícil de aprender
PEOU5	Pienso que sería fácil implementar una instancia de esta arquitectura.
PU1	Creo que se reduciría el tiempo y el esfuerzo requerido si se necesitara diseñar una solución AAL siguiendo arquitectura propuesta
PU2	Creo que es útil la forma en la que se describen cada uno de los componentes de la arquitectura
PU3	Creo que la arquitectura (sus elementos y componentes) NO es lo suficientemente expresiva para definir un Ambiente de Vida Asistido
PU4	El uso de esta arquitectura mejoraría mi rendimiento en la implementación de la misma

PU5	De manera general, pienso que con esta arquitectura NO se integran adecuadamente los componentes en la arquitectura CloudIoT
PU6	De manera general, considero que la arquitectura y sus componentes son útiles al momento de plantear una solución
ITU1	Si tuviera que utilizar una arquitectura para una solución IoT CloudIoT en el futuro, creo que tendría en cuenta esta.
ITU2	En caso de necesitar adaptar esta arquitectura CloudIoT a un escenario similar, tendría la intención de usarla en el futuro.
ITU3	No recomendaría el uso de esta arquitectura

Tabla 6-1 Ítems definidos para medir las variables basadas en la percepción. Fuente: Elaboración propia.

6.4 Evaluando la utilidad percibida de la Arquitectura CloudIoT la práctica: cuasi-experimento.

En esta sección, se presenta un cuasi-experimento llevado a cabo para validar empíricamente la fase de la modificación de la Arquitectura CloudIoT propuesta.

Un cuasi-experimento es una investigación empírica, similar a un experimento, en la cual la asignación de tratamiento a sujetos no puede estar basada en la aleatoriedad, pero emerge desde las características de los sujetos u objetos en sí mismo (C. Wohlin, 2007).

Las directrices y metodología seguidas en esta sección han sido tomadas del libro *Experimentation in Software Engineering* (Claes Wohlin et al., 2012) dirigido a investigadores y lectores que estudian informes de estudios realizados. Los lineamientos han sido adaptados para aplicarla al cuasi-experimento.

Se eligió un diseño de cuasi-experimento fijo y explicatorio (Runeson, Höst, Rainer y Regnell, 2012) con encuestas y cuestionarios como método de recopilación de datos. El proceso consta de las siguientes cinco fases interrelacionadas (consulte la Figura 6-5 para obtener una descripción general, incluidas las entradas y salidas de las diferentes fases):

1. Definición de objetivos y preguntas de investigación
2. Diseño y planificación, incluidos los preparativos para las pruebas y cuestionarios.
3. Recolección de evidencia (mediante los cuestionarios y encuestas)
4. Análisis de datos (calificación, agrupamiento, interpretación)
5. Informe

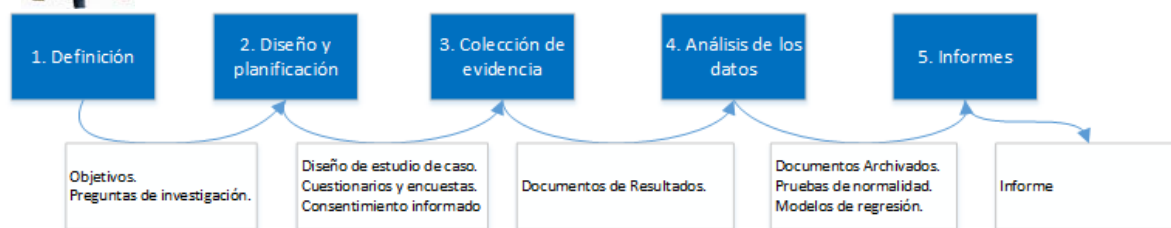


Figura 6-5 Descripción general del proceso de investigación que incluye entradas y salidas para cada fase. Fuente: Elaboración propia.

Las fases 1-4 se presentan con más detalle en las Secciones 6.4.1 a 6.4.4, mientras que las amenazas a la validez se discuten en la Sección 6.4.5. Es informe corresponde a la documentación correspondiente a cada fase realizada para el diseño del cuasi-experimento, es decir, el informe corresponde a toda la sección 6.4.

6.4.1 Definición de objetivos y preguntas de investigación

Esta fase inicial (ver Fig. 5-3) proporcionó la dirección y el alcance para el resto del estudio de caso. Se definieron un conjunto de objetivos y preguntas de investigación en base a la experiencia previa, los resultados y el conocimiento de los investigadores participantes, y un estudio de literatura en el área (Capítulo 3 Estado del Arte).

De acuerdo al paradigma Goal-Question Metric (GQM) propuesta Basili, Caldiera y Rombach (2002) la meta u objetivo de este experimento ha sido definida de la siguiente manera:

Evaluar	La fase de entendimiento y modificación de la Arquitectura CloudIoT
Con el propósito de	Evaluar la arquitectura con respecto a su eficacia percibida.
Desde el punto de vista del Contexto	Investigador Grupo de personas en período de formación en el campo de Ciencias de la Computación y de la Ingeniería de Software.

Tabla 6-2 Meta del experimento de acuerdo al paradigma Goal-Question Metric (GQM). Fuente: Elaboración propia.

En vista de que actualmente no existe un estándar para generar y modificar arquitecturas CloudIoT ampliamente aceptado, nosotros no podemos evaluar la Arquitectura CloudIoT con respecto a otra que sirva de control. Por esta razón,

se ha decidido llevar a cabo un cuasi-experimento con el objetivo de evaluar la arquitectura y probarla empíricamente.

Las preguntas planteadas en la investigación son:

RQ1: ¿La Arquitectura CloudIoT es percibida como fácil de usar y útil? De ser así, ¿las percepciones de los usuarios son el resultado de su rendimiento cuando utilizan la arquitectura y realizan modificaciones sobre ella?

RQ2: ¿Existe una intención de uso de la Arquitectura CloudIoT en el futuro? De ser así, ¿tales intenciones de uso es el resultado de las percepciones de los participantes?

La primera pregunta de investigación puede ser estudiada mediante las siguientes hipótesis:

- $H1_0$: La arquitectura CloudIoT se percibe como difícil de usar, $H1_0 = \neg H1_1$
- $H2_0$: La arquitectura CloudIoT no se percibe como útil, $H2_0 = \neg H2_1$
- $H4_0$: La facilidad de uso percibida no puede verse determinada por la eficiencia, $H4_0 = \neg H4_1$
- $H5_0$: La percepción de la utilidad no está determinada por la efectividad, $H5_0 = \neg H5_1$

La segunda pregunta de investigación puede ser estudiada a través de las siguientes hipótesis:

- $H3_0$: No existe intención de utilizar esta arquitectura CloudIoT en el futuro, $H3_0 = \neg H3_1$
- $H6_0$: La utilidad percibida no es determinada por la facilidad de uso percibida, $H6_0 = \neg H6_1$
- $H7_0$: La intención de uso no es determinada por la facilidad de uso percibida, $H7_0 = \neg H7_1$
- $H8_0$: La intención de uso no está determinada por la utilidad percibida, $H8_0 = \neg H8_1$

6.4.2 Diseño y Planificación del cuasi-experimento

En esta fase, se diseñaron los procedimientos de investigación y se hicieron preparativos para la recopilación de datos. Estas preparaciones incluyeron el diseño de las tareas, material experimental, la selección del escenario y los participantes.

El cuasi-experimento aplicado a la arquitectura propuesta se realizó como parte del proyecto “Evaluación del conocimiento sobre la alimentación, ejercicio, redes sociales y desarrollo de prototipos para la toma de medicación y monitorización de signos vitales orientados al adulto mayor mediante la integración y el uso de Tecnologías de la Información y Comunicación, aplicado a personas mayores de 65 años”. El proyecto tubo como parte de su investigación el análisis de diferentes propuestas para el control y asistencia médica de adultos mayores en cuanto la toma de medicación y control de signos vitales en entornos de vida asistido. Un enfoque particular de la investigación fue el diseño de arquitecturas que integren los paradigmas de IoT y Computación en la Nube.

6.4.2.1. Selección del contexto

El contexto está determinado por la arquitectura a ser evaluada, la selección de las modificaciones a ser realizadas y la selección de los participantes.

La arquitectura a ser evaluada es la presentada en este trabajo, concentrándose en la actividad de modificación, la cual es usada para agregar, quitar o modificar componentes en la arquitectura.

Esta actividad es importante debido a la necesidad de la interacción del usuario con la arquitectura CloudIoT. Los sujetos, en este caso actúan asumiendo el rol de arquitectos de software, el cual incluye las siguientes tareas: (i) entender el dominio de la arquitectura, (ii) identificar cada componente de la arquitectura, (iii) seleccionar los componentes que necesitan ser modificados, y (iv) generar la nueva arquitectura.

La arquitectura a ser evaluada pertenece al dominio de Ambient Assisted Living (AAL) y está descrita en el capítulo 5 más detalladamente con cada uno de sus componentes.

Para realizar el experimento, 30 participantes fueron seleccionados, todos ellos estudiantes de la Universidad de Cuenca. Los estudiantes tienen un muy buen conocimiento realizado a arquitecturas de software.

6.4.2.2. Tareas experimentales

El cuasi-experimento consistió en una tarea con cuatro sub-tareas:

Tarea: La modificación de componentes en una arquitectura CloudIoT y en base a ello la generación una arquitectura CloudIoT nueva. Para cada subtarea, los participantes tienen que entender el componente que se desea modificar y en base a ello fueron generando la nueva arquitectura.

- Subtarea 1: La integración de la arquitectura CloudIoT con un sistema externo. Para ello los participantes tuvieron que usar el componente adecuado para integrar el sistema y finalmente comunicar el sistema externo utilizando protocolos adecuados de comunicación.
- Subtarea 2: La incorporación de un nuevo dispositivo electrónico en un módulo de la arquitectura. En esta subtarea, los participantes tienen que agregar subcaracterísticas de seguridad en el dispositivo y acoplar de la mejor manera los componentes.
- Subtarea 3: La incorporación de paneles de monitoreo en lugar de los existentes; para ello cada panel de monitoreo tiene su propio protocolo de comunicación.
- Subtarea 4: La corrección de un error en uno de los puntos anteriores. Esto es necesario para verificar si los participantes entendieron correctamente la arquitectura y qué tan rápido se pueden realizar cambios en la arquitectura en caso de darse un error.

6.4.2.3 Variables

La Tabla 6-3 muestra las variables dependientes de interés basadas en la percepción, de acuerdo al MEM, las cuales fueron usadas para evaluar la arquitectura CloudIoT en la práctica.

Variables	Descripción
Facilidad de Uso Percibida (PEOU)	El grado en el cual los participantes creen que al usar la Arquitectura CloudIoT estarán libres de esfuerzo.
Utilidad Percibida (PU)	El grado en el cual los participantes creen que usando la Arquitectura CloudIoT se incrementará su rendimiento.
Intención de Uso (ITU)	El grado en el cual piensan usar la Arquitectura CloudIoT de necesitar una arquitectura CloudIoT en el dominio de AAL. Esto representa un juicio de la eficacia del método y puede ser utilizado para predecir la aceptación de la arquitectura en la práctica.

Tabla 6-3 Variables dependientes basadas en la percepción. Fuente: Elaboración propia.

Estas variables son medidas usando un cuestionario con una escala de Likert con un conjunto de 14 preguntas cerradas. El valor agregado para cada

variable subjetiva fue calculado como la media aritmética de las respuestas a las preguntas asociadas con cada variable dependiente subjetiva.

En la Tabla 6-4 se muestran las variables basadas en el rendimiento de interés y la función de medición usada para determinar sus valores.

Variable	Descripción
Efectividad	$\frac{\sum_{i=1}^n Tarea_i \text{ ejecutada correctamente}}{n}$
Eficiencia	$\sum_{i=1}^n Tiempo \text{ ejecutando la tarea}_i$

Tabla 6-4 Variables dependientes basadas en el rendimiento. Fuente: Elaboración propia.

6.4.2.4 Material experimental

El material experimental se compone del conjunto de documentos que son necesarios para realizar las tareas experimentales y el cuestionario para medir la percepción del usuario una vez que se ha realizado el experimento.

Este material incluye todo lo referente a las presentaciones con los conceptos, explicación de tareas y ejemplos que son utilizadas para el entrenamiento de los participantes.

La documentación utilizada, en su totalidad ha sido incluida en los anexos de este trabajo y es explicada a continuación.

1. Un documento que contiene la descripción del escenario sobre el que ha sido construida la arquitectura y la descripción de cada una de las subtareas. Se solicitó a los participantes que escriban la hora exacta de inicio y fin de la tarea a resolver.
2. Un anexo detallado como soporte, el cual contiene el metamodelo de la arquitectura con sus tipos de datos.
3. Una hoja con la arquitectura original.
4. Una hoja para dibujar la arquitectura luego de las modificaciones.
5. El cuestionario, el cual está disponible on-line.

En cuanto a las relaciones éticas del cuasi-experimento, este fue sujeto a un acuerdo de confidencialidad entre los participantes y los investigadores. En el estudio AAL, todos los registros fueron guardados en un servidor en la nube con acceso único de los investigadores. Al comienzo de cada prueba, obtuvimos el consentimiento individual de cada uno entrevistado a través de una carta de consentimiento que se encuentra en el Anexo A: Consentimiento



informado. Ningún entrevistado expresó inquietudes después de comprender la implicación de la carta de consentimiento.

6.4.3 Colección de la evidencia del cuasi-experimento

Antes de la sesión experimental se realizó un entrenamiento de 30 minutos con el objetivo de presentar los conceptos de computación en la nube, internet de las cosas y arquitecturas CloudIoT.

El entrenamiento incluyó el uso de la arquitectura y realizar modificaciones en la misma, incluyendo tareas envueltas en este proceso. El cuasi-experimento se realizó con un grupo de 30 estudiantes de Ingeniería de Sistemas en la Universidad de Cuenca.

La ejecución del cuasi-experimento fue controlada, no existieron interacciones significativas entre participantes. El cuasi-experimento fue conducido en una sesión de 60 minutos, incluido el entrenamiento. Los participantes que realizaron el experimento clarificaron sus dudas a lo largo de la sesión de entrenamiento. Después de las tareas experimentales, los participantes llenaron el cuestionario mostrado en la Tabla 6-1.

6.4.4 Análisis de los datos del experimento

Para el análisis de los resultados, se usaron pruebas, estadística descriptiva y box plots para analizar los datos recogidos en la fase anterior. Los datos estadísticos fueron obtenidos usando SPSS v20.

Una vez que los datos se recopilaron a través del material experimental (ver Fig. 6-5), se realizó un proceso de análisis en tres etapas que consistió en: análisis de las percepciones del usuario, análisis del rendimiento y análisis de las relaciones causales.

6.4.4.1 Análisis de las Percepciones de Usuario

La figura 6-6 muestra los diagramas de caja para cada variable de percepción en los cuales podemos ver que la media para cada variable es mayor que el valor neutro de la escala de Likert, que es el 3.

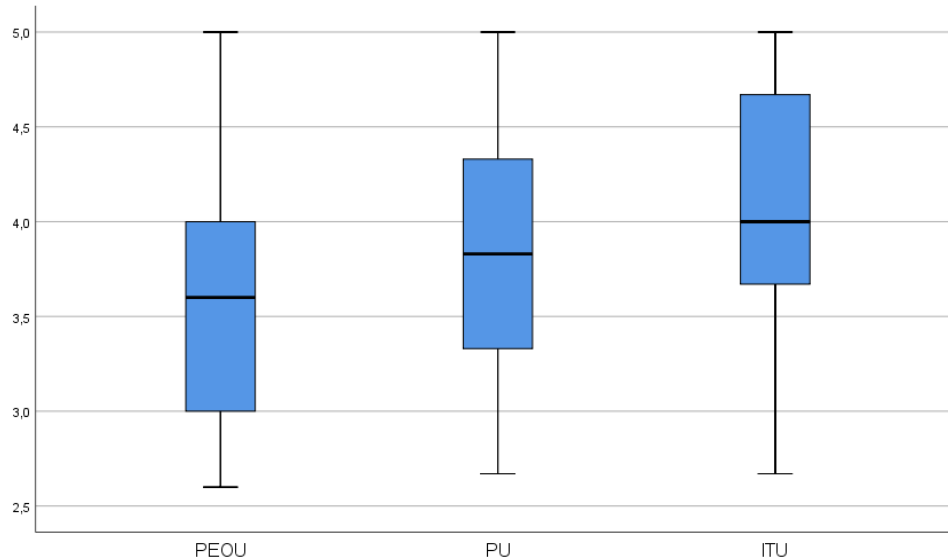


Figura 6-6 Diagrama de cajas para las variables PEOU, PU e ITU. Cuasi-Experimento. Fuente: Elaboración propia.

Se ha aplicado la prueba de Shapiro-Wilk para comprobar si los datos estaban distribuidos normalmente para seleccionar qué test podría usarse para chequear las hipótesis H1, H2 y H3.

Las tablas 6-5 y 6-6 muestran los resultados de la prueba Shapiro-Wilk y prueba t, respectivamente, para las variables estudiadas. Se aplicaron pruebas para verificar la hipótesis comparando si la media de las respuestas a las preguntas relacionadas con una variable dada, fueron significativamente más altas que el valor neutro de la escala Likert.

Ya que las pruebas de normalidad para las variables PEOU, PU e ITU tienen una distribución normal ($p > 0.05$) se ha probado la hipótesis aplicando el t-test one-tailed. Esos resultados nos permiten rechazar las hipótesis nulas $H1_0$, $H2_0$ y $H3_0$, lo que significa que los participantes perciben que la arquitectura CloudIoT es fácil de usar, útil y que ellos considerarían este método en el futuro si tuvieran que utilizar arquitecturas CloudIoT.

Variables	Valor neutro	Shapiro-Wilk		
		W	df	sig.
PEOU	3	,953	30	,199
PU	3	,941	30	,094
ITU	3	,948	30	,153

Tabla 6-5 Prueba de Shapiro-Wilk para las variables subjetivas. Fuente: Elaboración propia.

Valor de prueba = 3

	t	gl	Significancia (bilateral)	Diferencia de medias	95% de intervalo de confianza de la diferencia	
					Inferior	Superior
PEOU	5,508	29	<0,001**	,66000	,4149	,9051
PU	6,722	29	<0,001**	,90067	,6266	1,1747
ITU	8,918	29	<0,001**	1,04500	,8053	1,2847

Tabla 6-6 Prueba t con una muestra para las variables subjetivas. Fuente: Elaboración propia.

6.4.4.2 Análisis del Rendimiento del Usuario

Se ha medido la efectividad y la eficiencia de los participantes cuando utilizan esta arquitectura CloudIoT en la práctica. La tabla 6-7 presenta los valores de estadística descriptiva para las variables basadas en el rendimiento. La efectividad total fue en promedio del 70.67%, indicando que la mayoría de los participantes fueron capaces de ejecutar las tareas correctamente.

La eficiencia ha sido calculada como el esfuerzo requerido para realizar las tareas (Moody, 2001). Los resultados muestran que la eficiencia de los participantes fue de 14 a 52 minutos.

	Mínimo	Máximo	Media	Desv. Desviación
EFICIENCIA	14	52	23,93	7,969
EFFECTIVIDAD	,20	1,00	,7067	,25587

Tabla 6-7 Estadística Descriptiva para Variables Basadas en la percepción del Usuario.

6.4.4.3 Análisis de las Relaciones Causales

El objetivo de esta sección es validar la parte estructural del MEM en términos de las relaciones causales entre sus constructores, con la excepción del Uso Actual. Para esto, se ha utilizado análisis de regresión para evaluar la operacionalización del MEM realizada, ya que las hipótesis a ser probadas son relaciones causales entre variables continuas. Para realizar este análisis, hemos utilizado los siguientes niveles de significancia sugeridos por Moody (Moody, 2001):

Valor de Significancia	Rango
No significativo	$p > 0.1$
Baja significancia	$p < 0.1$
Media significancia	$p < 0.05$
Alta significancia	$p < 0.01$
Muy alta significancia	$p < 0.001$

Tabla 6-8 Niveles de significancia. Fuente: Moody (2001)

6.4.4.4 Eficiencia vs Facilidad de Uso Percibida

La hipótesis H4 ha sido probada para comprobar si las percepciones de Facilidad de Uso Percibida (PEOU) son determinadas por la Eficiencia de los participantes cuando se aplica el método. Para ejecutar este análisis se ha construido un modelo de regresión simple en el cual la eficiencia fue usada como la variable independiente (predictora) y PEOU como la variable dependiente (predicha). La ecuación de regresión resultante del análisis es la siguiente:

$$PEOU = 3.827 + (-0.085) * Eficiencia$$

	B	Desv. Error	Beta	t	Sig.	R	R cuadrado
(Constante)	3,827	,391		9,801	,000		
EFICIENCIA	-,007	,016	-,085	-,451	,656	,09	,01

a. Variable dependiente: PEOU

Tabla 6-9 Regresión Simple entre la Eficiencia Actual y la Facilidad de Uso Percibido. Fuente: Elaboración propia.

El modelo de regresión fue encontrado no significativo, con $p > 0.1$. El R cuadrado muestra que la variable eficiencia permite explicar solamente el 1% de la varianza PEOU, indicando que la eficiencia actual de los participantes no influencia sus percepciones de facilidad de uso (ver Tabla 6-9). Estos resultados no nos permiten rechazar la hipótesis nula $H4_0$ y aceptar su hipótesis alternativa, significando que hemos corroborado que la facilidad de uso percibido (PEOU) no está determinada por la Eficiencia.

6.4.4.5 Efectividad vs Utilidad Percibida

La hipótesis H5 ha sido probada para verificar si las percepciones de la Utilidad Percibida (PU) están determinadas por la Efectividad de los participantes. Similarmente, nosotros construimos un modelo de regresión simple en el cual la Efectividad fue usada como la variable independiente mientras que la PU fue usada como variable dependiente. La ecuación obtenida desde el modelo es la siguiente.

$$PU = 3.137 + 1.081 * Efectividad$$

	B	Desv. Error	Beta	t	Sig.	R	R cuadrado
(Constante)	3,137	,377		8,329	,000		
EFFECTIVIDAD	1,081	,502	,377	2,152	,040	,38	,14

a. Variable dependiente: PU

Tabla 6-10 Regresión simple entre la Efectividad Actual y la Utilidad Percibida. Fuente:
Elaboración propia.

El modelo de regresión presenta una significancia media con $p < 0.05$. El R cuadrado muestra que la Efectividad es capaz de explicar el 14% de varianza de PU, indicando que ciertas percepciones con respecto a PU están determinadas por la efectividad de los participantes cuando realizan las tareas sobre la arquitectura CloudIoT (ver Tabla 6-10). Como se esperaba el coeficiente de regresión para la efectividad fue positivo, lo que significa que a valor más alto para la efectividad más alto el valor de la Utilidad Percibida. Esos resultados nos permiten rechazar la hipótesis nula $H5_0$ y aceptar su hipótesis alternativa, lo que significa que empíricamente se ha probado que PU está determinado por la efectividad.

6.4.4.6 PEOU vs Utilidad Percibida

La hipótesis H6 ha sido probada para verificar si las percepciones de la Utilidad Percibida (PU) están determinadas por la Facilidad de Uso Percibida (PEOU). Similarmente, se construyó un modelo de regresión en el cual la variable PEOU fue usada como variable independiente mientras que PU fue usada como la variable dependiente.

La ecuación obtenida del modelo de regresión es:

$$PU = 1.145 + 0.753 * PEOU$$

	B	Desv. Error	Beta	t	Sig.	R	R cuadrado
(Constante)	1,145	,581		1,971	,059		
PEOU	,753	,156	,673	4,820	<0.001	,67	,45

a. Variable dependiente: PU

Tabla 6-11 Regresión Simple entre la Facilidad de Uso Percibida y la Utilidad Percibida. Fuente:
Elaboración propia.

Se ha encontrado que el modelo de regresión es altamente significativo con $p < 0.001$. R cuadrado muestra que la variable Facilidad de Uso Percibida es capaz de explicar el 45% de la varianza en PU, indicando que ciertas percepciones con respecto a PU están determinadas por PEOU (ver Tabla 6-11). Estos resultados nos permiten rechazar $H6_0$ y aceptar su hipótesis alternativa, lo que significa que hemos corroborado empíricamente que PU está determinada por PEOU.



6.4.4.7 Intención de Uso vs Utilidad Percibida

La hipótesis H7 ha sido probada para verificar si las percepciones de la Intención de Uso (ITU) están actualmente determinadas por la utilidad percibida (PU) cuando se aplica el método. La ejecución de este análisis fue construida a partir de un modelo simple de regresión en el cual la variable PU fue usada como variable independiente mientras ITU fue usada como variable dependiente, la ecuación obtenida del modelo es como sigue:

$$ITU = 1.671 + 0.609 * PU$$

	R	R cuadrado	B	Desv. Error	Beta	t	Sig.
(Constante)			1,671	,471		3,549	,001
PU	,70	,48	,609	,119	,696	5,129	<0.001

a. Variable dependiente: ITU

Tabla 6-12 Regresión Simple entre Utilidad Percibida e Intención de Uso. Fuente: Elaboración propia.

Mediante el modelo de regresión se encontró una alta significancia, con $p < 0.001$. R cuadrado muestra que la variable PU es capaz de explicar el 48% de la varianza en ITU, lo cual representa un alto valor dado que podrían existir otros factores que influyeran la intención de los participantes de usar un método (ver Tabla 6-12). Estos resultados permiten rechazar $H7_0$ y aceptar su hipótesis alternativa, lo que significa que hemos corroborado que ITU está determinado por PU.

6.4.4.8 Intención de Uso vs. Facilidad de Uso Percibida

La hipótesis H8 ha sido probada para verificar si la Intención de Uso (ITU) está actualmente determinada por la Facilidad de Uso Percibida (PEOU). Similarmente, se construyó un modelo de regresión simple en el cual la variable PEOU fue usada como una variable independiente e ITU como una variable dependiente. La ecuación obtenida desde el modelo es la siguiente:

$$ITU = 2.292 + 0.479 * PEOU$$

	R	R cuadrado	B	Desv. Error	Beta	t	Sig.
(Constante)			2,292	,599		3,828	,001
PEOU	,49	,24	,479	,161	,490	2,973	,006



a. Variable dependiente: ITU

Tabla 6-13 Regresión Simple entre Facilidad de Uso Percibida e Intención de Uso. Fuente:
Elaboración propia.

A través de este análisis se encontró una alta significancia con $p < 0.01$. El R cuadrado muestra que la variable PEOU permite explicar el 24% de la varianza en ITU, indicando que las intenciones de los participantes para usar el método en el futuro están determinadas en cierta medida por su percepción en la facilidad de uso del método (ver Tabla 6-13). Estos resultados permiten rechazar H_{8_0} y aceptar la hipótesis alternativa, lo que significa que se ha corroborado empíricamente que ITU está determinada por PEOU.

6.5 Amenazas a la validez

En esta sección, se explican los principales problemas que pueden poner en peligro la validez del cuasi-experimento, al considerar los cuatro tipos de amenazas que proponen Cook y Campbell (1979).

6.5.1 Validez interna

Las principales amenazas para la validez interna fueron: la experiencia de los participantes, los sesgos del autor y los sesgos relacionados con la arquitectura CloudIoT.

Para reducir la amenaza con respecto a la experiencia de los participantes, se ha preparado un ejemplo de entrenamiento representativo, el cual muestra los pasos para entender la arquitectura y provee a los usuarios un alto entendimiento en arquitecturas CloudIoT.

Tanto los sesgos del autor como los producidos por la entendibilidad del material fueron reducidos durante la ejecución de un experimento piloto, en el cual una investigadora experta en el dominio evaluó el material para así reducir posibles errores o malos entendidos relacionados con el experimento.

Los sesgos relacionados a la arquitectura fueron reducidos a través de su validación en el experimento piloto y pruebas sucesivas para mejorar el manejo de la arquitectura.

6.5.2 Validez externa

Esta se refiere a la habilidad para generalizar los resultados a diferentes contextos. La principal amenaza puede verse afectada por el diseño de la evaluación, el contexto de los participantes y el tamaño o complejidad de las tareas.

El diseño de la evaluación puede tener un impacto en la generalización de los resultados debido a la complejidad de la arquitectura, sus características

particulares, el dominio de la misma y los componentes que deberían ser considerados. Para reducir este problema, se ha intentado generalizar un escenario, de modo que éste pueda ser adaptado fácilmente a otro contexto similar. Además, se han seleccionado tareas sobre los componentes que son comunes en otros tipos de arquitecturas. El cuasi-experimento fue conducido con alumnos de Ingeniería de Sistemas, quienes tienen conocimientos sobre arquitecturas de software.

6.5.3 Validez del constructo

La principal amenaza en la validez del constructor es el cómo reflejar la eficacia en la manera en que la arquitectura CloudIoT ha sido analizada y modificada. Las variables subjetivas están basadas en el *Method Evaluation Model* (MEM) (Moody, 2001), el cual es un método conocido y validado empíricamente para la evaluación de tecnologías de la información. Por otro lado, para analizar la confiabilidad del cuestionario se ha realizado un análisis de la prueba de confiabilidad del alfa de Cronbach fue realizada para cada conjunto de preguntas relacionadas con cada variable subjetiva. Los resultados fueron, en la facilidad de uso percibido $\alpha = 0.795$ utilidad percibida es $\alpha = 0.861$ e intención de uso es $\alpha = 0.681$. Según, Bosma et al. (Bosma et al., 1997) el alfa de Cronbach aceptable para experimentos está en el intervalo $0.67 < \alpha < 0.95$.

6.5.3 Validez de la conclusión

Las amenazas que afectan a la validez de la conclusión se refieren a las conclusiones estadísticas. Ejemplos de estas son la elección de los métodos estadísticos, y la elección del tamaño de la muestra, entre otros.

Uno de los principales problemas de la validez es el tamaño de la muestra para realizar el cuasi-experimento; sin embargo, los resultados fueron alentadores, ya que los participantes lograron realizar con éxito las tareas propuestas.

Finalmente, para controlar el sesgo de la variación debido a diferencias individuales de los participantes, se ha seleccionado un grupo homogéneo de participantes, además del entrenamiento previo antes de realizar las tareas.

6.7 Conclusiones

En esta sección se ha presentado un cuasi-experimento, con lo cual se ha evaluado la eficacia percibida de un grupo de estudiantes utilizando la Arquitectura CloudIoT propuesta para realizar una serie de modificaciones y análisis sobre la misma.

Aunque los resultados son favorables en cuanto a la utilidad de la arquitectura para apoyar a los *usuarios* en entender y modificar la arquitectura CloudIoT



propuesta, es necesario realizar otros experimentos que consideren otros factores importantes.

Además, es necesario analizar la arquitectura en otros escenarios, ya que, si se realizan pequeños cambios sobre la misma, la evaluación cambiaría drásticamente.

Capítulo 7

Conclusiones y trabajos futuros

7.1 Conclusiones

A continuación, se detalla la consecución de cada uno de los objetivos que han sido inicialmente planteados.

7.1.1 Objetivo general

El objetivo general de esta tesis es *investigar y analizar los componentes que deben ser considerados al momento de combinar la Computación en la Nube (Cloud Computing) con el Internet de las cosas (IoT), a fin de proponer el diseño de una arquitectura tecnológica siguiendo los lineamientos del paradigma CloudIoT aplicándolo a un ambiente de tecnología asistiva.*

Este objetivo se ha cumplido en su totalidad, dado que, a lo largo de este trabajo, se ha planteado una arquitectura CloudIoT en un escenario de AAL que permite integrar componentes de Computación en la Nube e IoT. Además, se ha diseñado un DSM, que permite estandarizar la composición e integración entre elementos de la arquitectura, además de que permite extenderla en función de sus requisitos. Se ha realizado también un estudio empírico para proporcionar evidencia sobre la utilidad de la arquitectura y apoyar a los stakeholders en el diseño e implementación de arquitecturas CloudIoT en el dominio de AAL.

7.1.2 Objetivos específicos

Los objetivos específicos alcanzados se detallan a continuación.

7.1.2.1 Objetivo específico 1

Estudiar el estado del arte relacionado con la integración de las tecnologías de Computación en la Nube e IoT.

Este objetivo ha sido alcanzado mediante la investigación de diferentes arquitecturas que involucran tecnologías de Computación en la Nube, IoT y AAL.

Como se ha explicado detalladamente en el Capítulo 3, existen arquitecturas de Computación en la Nube e IoT, las cuales pueden ser aplicadas o se enfocan en ciertos dominios específicos, además algunas de ellas definen DSLs para abstraer ciertos componentes y lograr una heterogeneidad entre los mismos. Sin embargo, las investigaciones no se han enfocado específicamente a soluciones de arquitecturas CloudIoT en ambientes de vida asistidos (AAL).

Además, se ha observado que existe una gran variabilidad al momento de construir arquitecturas de Computación en la Nube e IoT; es decir, no existe una manera que permita estandarizar las arquitecturas ya que estas dependen mucho del tipo de usuario que hará uso de la solución y sus requisitos específicos, es por eso que una de las principales metas alcanzadas de esta tesis, ha sido el generalizar este tipo de arquitecturas mediante Mod4CloT, para que las arquitecturas necesarias puedan ser especificadas de acuerdo a una semántica apropiada y que puedan además ser construidas en base a requerimientos tanto funcionales como no funcionales.

7.1.2.2 Objetivo específico 2

Definir un lenguaje y un modelo de dominio específico (DSL, DSM) que permita representar una arquitectura CloudIoT.

Para cumplir este objetivo, se ha definido un metamodelo que integra los diferentes componentes necesarios para generar una arquitectura CloudIoT. Posteriormente, se ha generado un DSM con base en el metamodelo, con el cual se ha generado una arquitectura CloudIoT. Este DSM permite generar arquitecturas CloudIoT en distintos dominios y al tener el DSM, la arquitectura se genera semánticamente, de modo que tanto los desarrolladores, como los expertos en el dominio, pueden manejarla.

7.1.2.3 Objetivo específico 3

Proponer una arquitectura que integre las tecnologías de Computación en la Nube e IoT en campo de tecnología asistiva y aplicarlo a un escenario real.

Para cumplir este objetivo, se ha generado un caso de estudio, con un escenario de la vida real. En este escenario se han descrito los principales actores y componentes que deberían interactuar para cumplir con los requisitos del caso en el dominio de AAL.

Además, se han analizado cada uno de los componentes y posteriormente se ha realizado una evaluación de la arquitectura generada para demostrar empíricamente que esta arquitectura ha realizado un aporte en investigación.

7.1.2.3 Objetivo específico 4

Evaluar la solución en un dominio específico con la ayuda de expertos en el mismo.

Para cumplir este objetivo, se ha realizado un cuasi-experimento con personas que tienen conocimientos sobre arquitecturas de software. Aquí se ha analizado la eficacia de la arquitectura y se ha medido la percepción del usuario (la facilidad de uso percibida, utilidad percibida e intención de uso). Tras los resultados obtenidos, se ha encontrado que la arquitectura es fácil de usar y útil. Además, la mayoría de los participantes han expresado su intención de uso de esta arquitectura en el futuro.

7.1.3 Conclusiones generales

En este trabajo de titulación se ha construido una arquitectura CloudIoT en el dominio de AAL mediante el DSM Mod4CloT y uno de los mayores retos fue definir un modelo que cumpla con los requisitos necesarios en este dominio. Es por ello, que los componentes que se han generado tienen un alto nivel de abstracción para que sea entendible por expertos en el dominio y generalizar este tipo de arquitecturas mediante Mod4CloT, para que las

arquitecturas necesarias puedan ser construidas en base a los requerimientos de las partes interesadas o *skateholders*; además cada componente es generado semánticamente para que los desarrolladores puedan manejarlos y configurarlos.

Cada arquitectura cumple objetivos y requisitos diferentes, es por ello que mediante Mod4CloT hemos intentado abstraer los componentes de la arquitectura. Es decir, si se requieren agregar estándares de salud, estándares de la nube, estándares IoT o requisitos específicos de esa arquitectura, es posible hacerlo al agregar ciertos componentes como protocolos, requisitos de seguridad, adaptadores, entre otros.

7.2 Trabajo Futuro

Este trabajo de titulación no es el final de los esfuerzos de investigación en esta área. Muchas actividades aún se pueden realizar para mejorar y ampliar nuestra arquitectura propuesta y el DSM CloudIoT. Los principales aspectos que se piensa abordar son analizados a continuación.

- Con respecto al metamodelo, se piensa que realizar un análisis minucioso de todos los componentes que interactúan en aplicaciones que integran las tecnologías de Computación en la Nube e IoT, abarcando los problemas derivados de su integración (p. ej. seguridad, falta de estandarización, heterogeneidad, entre otros) sería necesario y daría más robustez a esta línea de investigación.
- Con respecto a la arquitectura propuesta, sería necesario mejorar el diseño en base a las sugerencias y comentarios encontrados en el experimento. De igual manera, en la arquitectura instanciada a partir del metamodelo, definir a profundidad los detalles de las plataformas o tecnologías con que se implementará la solución.
- También se espera como trabajo futuro, generar operaciones de mapeo entre la arquitectura instanciada y código.
- Con los resultados obtenidos tras la validación empírica realizada en el presente trabajo, se prevé la mejora de los instrumentos y métodos de evaluación de esta solución, a fin de eliminar todos los posibles problemas que se han dado durante las sesiones de experimentación.
- Se espera además realizar réplicas de los estudios empíricos teniendo como sujetos a profesionales del área involucrados en la industria y

otros sectores que permitan validar la solución desde diferentes puntos de vista de usuarios.

- Además, como trabajo futuro se plantea la necesidad de buscar arquitecturas que integren tecnologías para poder realizar comparativas y afinar el trabajo presentado en este estudio.
- Finalmente, se esperan realizar pruebas en escenarios con requisitos diferentes y que involucren estándares médicos como HL7 o DICOM.

REFERENCIAS BIBLIOGRÁFICAS

- Aazam, M., & Huh, E. (2014). Fog Computing and Smart Gateway Based Communication for Cloud of Things.
<https://doi.org/10.1109/FiCloud.2014.83>
- Abrahão, S., Insfran, E., Carsí, J. A., & Genero, M. (2011). Evaluating Requirements Methods based on User Perceptions: A Family of Experiments.
<https://doi.org/10.1016/j.ins.2011.04.005>
- Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things : A survey. *Computer Networks*, 54(15), 2787–2805.
<https://doi.org/10.1016/j.comnet.2010.05.010>
- B.B, P. R. B., Saluia, P., Sharma, N., Mittal, A., & Sharma, S. V. (2012). Cloud computing for Internet of Things & sensing based applications. *Sensing Technology (ICST), 2012 Sixth International Conference On*, 374–380.
<https://doi.org/10.1109/ICSensT.2012.6461705>



- Babu, S. M., Lakshmi, A. J., & Rao, B. T. (2015). A Study on Cloud based Internet of Things : CloudIoT, (Gcct).
- Basili, V., Caldiera, G., & Rombach, H. D. (2002). Goal Question Metric (GQM) Approach. *Encyclopedia of Software Engineering*, 2, 1–10.
<https://doi.org/10.1002/0471028959.sof142>
- Bass, L., Clements, P., & Kazman, R. (2003). *Software architecture in practice*. Addison-Wesley.
- Bézivin, J. (2004). In search of a basic principle for Model Driven Engineering. *Special Novatica Issue - UML and Model Engineering*, 5(2), 21–24.
- Bosma, H., Marmot, M. G., Hemingway, H., Nicholson, A. C., Brunner, E., & Stansfeld, S. A. (1997). Low job control and risk of coronary heart disease in Whitehall II (prospective cohort) study. *Bmj*, 314(7080), 558–565.
<https://doi.org/10.1136/bmj.314.7080.541>
- Botta, A., De Donato, W., Persico, V., & Pescapé, A. (2014). On the integration of cloud computing and internet of things. *Proceedings - 2014 International Conference on Future Internet of Things and Cloud, FiCloud 2014*, 23–30.
<https://doi.org/10.1109/FiCloud.2014.14>
- Botta, A., Donato, W. De, Persico, V., & Pescapé, A. (2016). Integration of Cloud computing and Internet of Things : A survey. *Future Generation Computer Systems*, 56, 684–700. <https://doi.org/10.1016/j.future.2015.09.021>
- Cedillo Orellana, I. P., Insfrán Pelozo, E., & Abrahao, S. (2017). Evaluación de un Método de Monitorización de Calidad de Servicios Cloud: Una Replicación Interna. Retrieved August 20, 2018, from <http://issi.dsic.upv.es/publications/articles?view=488>
- Cedillo, P. (2016). Monitorización de calidad de servicios cloud mediante modelos en tiempo de ejecución, (January), 1–361.
- Cervantes, H., & Kazman, R. (2016). *Designing software architectures : a practical approach*. Addison-Wesley Professional. Retrieved from <https://www.safaribooksonline.com/library/view/designing-software-architectures/9780134390857/>
- Chatzigiannakis, I., Hasemann, H., Karnstedt, M., Kleine, O., Kr??ller, A., Leggieri, M., ... Truong, C. (2012). True self-configuration for the IoT. *Proceedings of 2012 International Conference on the Internet of Things, IOT 2012*, 9–15.
<https://doi.org/10.1109/IOT.2012.6402298>
- CISCO. (2015). *Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are What You Will Learn*. Retrieved from https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf



- Cook, T. D., & Campbell, D. T. (1979). *Quasi-Experimentation: Design and Analysis Issues for Field Settings*.
- Cubo, J., Nieto, A., & Pimentel, E. (2014). *A Cloud-Based Internet of Things Platform for Ambient Assisted Living*. <https://doi.org/10.3390/s140814070>
- Dash, S. K., Mohapatra, S., & Pattnaik, P. K. (2010). A Survey on Applications of Wireless Sensor Network Using Cloud Computing. *International Journal of Computer Science & Emerging Technologies (E)*, 1(4), 2044–6004. Retrieved from <http://ijcset.excelingtech.co.uk/vol1issue4/10-vol1issue4.pdf>
- Davis, F. D. (1985). A technology acceptance model for empirically testing new end-user information systems : theory and results. Retrieved from <https://dspace.mit.edu/handle/1721.1/15192>
- Deursen, A. Van. (1997). Domain-Specific Languages versus Object-Oriented Frameworks : A Financial Engineering Case Study.
- Deursen, A. Van, & Klint, P. (1998). Little Languages : Little Maintenance? *Journal OfSoftware Maintenance*, 1–19.
- Deursen, A. Van, Visser, J., & Klint, P. (2000). Domain-Specific Languages : An Annotated Bibliography.
- Distefano, S., Merlino, G., & Puliafito, A. (2013). Towards the Cloud of Things Sensing and Actuation as a Service , a key enabler for a new Cloud paradigm. <https://doi.org/10.1109/3PGCIC.2013.16>
- Doukas, C., & Maglogiannis, I. (2012). Bringing IoT and Cloud Computing towards Pervasive Healthcare. <https://doi.org/10.1109/IMIS.2012.26>
- Doukas, C., Pliakas, T., Tsanakas, P., & Maglogiannis, I. (2012). *Distributed Management of Pervasive Healthcare Data through Cloud Computing. Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering* (Vol. 83). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-29734-2_53
- Eleftherakis, G., Pappas, D., Lagkas, T., Rousis, K., & Paunovski, O. (2015). Architecting the IoT Paradigm: A Middleware for Autonomous Distributed Sensor Networks. *International Journal of Distributed Sensor Networks*, 2015(December). <https://doi.org/10.1155/2015/139735>
- European Commission. (2013). *Definition of a Research and Innovation Policy Leveraging Cloud Computing and IoT Combination*. European Commission. <https://doi.org/10.2759/38400>
- Farahzadi, A., Shams, P., Rezazadeh, J., & Farahbakhsh, R. (2017). Middleware technologies for cloud of things-a survey. *Digital Communications and Networks*. <https://doi.org/10.1016/J.DCAN.2017.04.005>



- Fariba Sadri. (2011). *Ambient Intelligence : A Survey*, 43(4).
<https://doi.org/10.1145/1978802.1978815>
- Fishbein, M., & Ajzen, I. (1975). *Belief, attitude, intention, and behavior : an introduction to theory and research*. Addison-Wesley Pub. Co. Retrieved from https://books.google.com.ec/books?id=8o0QAQAIAAJ&q=Belief,+Attitude,+Intention+and+Behaviour:+An+Introduction+to+Theory+and+Research.&dq=Belief,+Attitude,+Intention+and+Behaviour:+An+Introduction+to+Theory+and+Research.&hl=es&sa=X&ved=0ahUKEwjy3_VkbbcAhUQw1kKHZg0CRUQ6AEIKDAA
- Fowler, M. (2010). *Domain-Specific Languages*. (Pearson Education, Ed.). Retrieved from <https://www.pearson.com/us/higher-education/program/Fowler-Domain-Specific-Languages/PGM305379.html>
- France, R., Rumpe, B., France, R., & Rumpe, B. (2007). *Model-driven Development of Complex Software : A Research Roadmap*.
- Garçon, L., Khasnabis, C., Walker, L., Nakatani, Y., Lapitan, J., Borg, J., ... Berumen, A. V. (2016). Medical and assistive health technology: Meeting the needs of aging populations. *Gerontologist*, 56, S293–S302.
<https://doi.org/10.1093/geront/gnw005>
- Gomberg, P., & Rescher, N. (1975). The Primacy of Practice. *The Philosophical Review*, 84(4), 603. <https://doi.org/10.2307/2183865>
- Green, J. (2014). The Internet of Things Reference Model. *Internet of Things World Forum*, 1–12.
- Hassan, M. M., Albakr, H. S., & Al-dossari, H. (2014). A Cloud-Assisted Internet of Things Framework for Pervasive Healthcare in Smart City Environment, 9–13.
- Health Organization World. (2017). Improving access to assistive technology. Retrieved from <http://www.who.int/disabilities/publications/technology/wheelchairguidelines/en/>,
- Hernández, R., Fernández, C., & Baptista, M. del P. (2014). *Metodología de la Investigación*.
- Hudak, P., & Cam, C. A. D. (1998). *Domain Specific Languages* *, III, 39–60.
- ISO/IEC. (2011a). Characteristics Sub-Characteristics, 2. Retrieved from https://edisciplinas.usp.br/pluginfile.php/294901/mod_resource/content/1/ISO_25010_-_Quality_Model.pdf
- ISO/IEC. (2011b). ISO/IEC 25010:2011(en) Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. Retrieved from



<https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en>

- ITU. (2005). The Internet of Things. *Itu Internet Report 2005*, 212.
<https://doi.org/10.2139/ssrn.2324902>
- Jadeja, Y. (2012). Cloud Computing - Concepts , Architecture and Challenges, 877–880.
- Jing, X., & Jian-jun, Z. (2010). A Brief Survey on the Security Model of Cloud Computing. *2010 Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science*, 475–478.
<https://doi.org/10.1109/DCABES.2010.103>
- Kuo, A. M.-H. (2011). Opportunities and Challenges of Cloud Computing to Improve Health Care Services. *Journal of Medical Internet Research*, 13(3), e67.
<https://doi.org/10.2196/jmir.1867>
- Kurtev, I. (2005). *Adaptability of Model Transformations. Work*.
- Leach, P. J., Berners-Lee, T., Mogul, J. C., Masinter, L., Fielding, R. T., & Gettys, J. (1999). Hypertext Transfer Protocol -- HTTP/1.1. Retrieved from
<https://tools.ietf.org/html/rfc2616>
- Lee, K., Murray, D., Hughes, D., Joosen, W., Lee, K., & Murray, D. (2010). Extending Sensor Networks into the Cloud using Amazon Web Services. *2010 IEEE International Conference on Networked Embedded Systems for Enterprise Applications*. <https://doi.org/10.1109/NESEA.2010.5678063>
- Leimeister, S., Böhm, M., München, T. U., Riedl, C., & München, T. U. (2010). The Business Perspective of Cloud Computing : Actors , Roles and Value Networks.
- Liddle, S. W. (2011). Model-Driven Software Development. *Handbook of Conceptual Modeling: Theory, Practice, and Research Challenges*, (Section 2), 17–54.
https://doi.org/10.1007/978-3-642-15865-0_2
- Lin, Y., Gray, J., & Jouault, F. (2007). DSMDiff: A differentiation tool for domain-specific models. *European Journal of Information Systems*, 16(4), 349–361.
<https://doi.org/10.1057/palgrave.ejis.3000685>
- Masse, M. (2011). *REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces*. (I. O'Reilly Media, Ed.).
- Mell, P., Grance, T., & Grance, T. (2011). The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology.
- Miao Yun, & Bu Yuxin. (2010). Research on the architecture and key technology of Internet of Things (IoT) applied on smart grid. In *2010 International Conference on Advances in Energy Engineering* (pp. 69–72). IEEE.
<https://doi.org/10.1109/ICAEE.2010.5557611>



- Microsoft. (2018). Architecture styles | Microsoft Docs. Retrieved August 1, 2018, from <https://docs.microsoft.com/es-es/azure/architecture/guide/architecture-styles/>
- Microsoft Patterns & Practices Team. (2009). *.NET application architecture guide*. Microsoft. Retrieved from <https://www.intertech.com/Downloads/eBook/ApplicationArchitectureGuide.pdf>
- Minerva, R., Abyi, B., & Rotondi, D. (2015). Towards a definition of the Internet of Things (IoT), 1–86.
- Ministerio de Salud Pública del Ecuador. (2012). *Manual del Modelo de Atención Integral de Salud - MAIS*. Retrieved from http://instituciones.msp.gob.ec/somossalud/images/documentos/guia/Manual_MAIS-MSP12.12.12.pdf
- Mohammed, J., Thakral, A., Ocneanu, A. F., Jones, C., Lung, C.-H., & Adler, A. (2014). Internet of Things : Remote Patient Monitoring Using Web Services and Cloud Computing, (iThings). <https://doi.org/10.1109/iThings.2014.45>
- Moody, D. (2001). *A Practical Method for Representing Large Entity Relationship Models*.
- MQTT.org. (2017). MQTT. Retrieved February 7, 2018, from <http://mqtt.org>
- OMG. (2008). Software & Systems Process Engineering Meta-Model Specification. Retrieved August 1, 2018, from <https://www.omg.org/spec/SPEM/About-SPEM/>
- Parra, J. M., Valdez, W. F., Guevara, A. P., Cedillo, I. P., & Ortiz-Segarra, J. (2017). Intelligent PillBox: Automatic and Programmable Assistive Technology Device. In *Biomedical Engineering*. Calgary, AB, Canada: ACTAPRESS. <https://doi.org/10.2316/P.2017.852-051>
- Perisic, B., & Sad, N. (2014). Model Driven Software Development – State of The Art and Perspectives, 13(March), 1237–1248.
- Prati, A., Vezzani, R., Fornaciari, M., & Cucchiara, R. (2013). Intelligent Video Surveillance as a Service. In *Intelligent Multimedia Surveillance* (pp. 1–16). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-41512-8_1
- Puliafito, A. (2014). SensorCloud : An Integrated System for Advanced Multi-risk Management. <https://doi.org/10.1109/NCCA.2014.10>
- Rashidi, P., & Mihailidis, A. (2013). A Survey on Ambient-Assisted Living Tools for Older Adults, 17(3), 579–590.



- Riemenschneider, C. K., Hardgrave, B. C., & Davis, F. D. (2002). Explaining software developer acceptance of methodologies: a comparison of five theoretical models. *IEEE Transactions on Software Engineering*, 28(12), 1135–1145. <https://doi.org/10.1109/TSE.2002.1158287>
- Runeson, P., Höst, M., Rainer, A., & Regnell, B. (2012). *CASE STUDY RESEARCH IN SOFTWARE ENGINEERING: Guidelines and Examples*. John Wiley & Sons. Retrieved from <http://www.egov.ee/media/1267/case-study-research-in-software-engineering.pdf>
- Salihbegovic, A., Eterovic, T., Kaljic, E., & Ribic, S. (2015). Design of a domain specific language and IDE for Internet of things applications, (May), 25–29.
- Schmidt, D. C. (2006). Model Driven Engineering, (February), 25–31.
- Shelke, Y., & Sharma, A. (2016). *INTERNET OF MEDICAL THINGS*. Retrieved from <https://www.aranca.com/knowledge-library/special-reports/ip-research/the-internet-of-medical-things-iomt>
- Steg, H., Strese, H., Hull, J., & Schmidt, S. (2005). Ambient Assisted Living Offers Solutions. In *Ambient Assisted Living. Europe Is Facing a Demographic Challenge* (pp. 238–248). European Overview Report. Retrieved from <https://www.forschungsinformationssystem.de/servlet/is/217135/>
- Steinberg, D., Budinsky, F., Merks, E., & Paternostro, M. (2008). *EMF: Eclipse Modeling Framework (Eclipse Series)*. Addison-Wesley Professional.
- Suciu, G., Vulpe, A., Halunga, S., Fratu, O., Todoran, G., & Suciu, V. (2013). Smart Cities Built on Resilient Cloud Computing and Secure Internet of Things. In *2013 19th International Conference on Control Systems and Computer Science* (pp. 513–518). IEEE. <https://doi.org/10.1109/CSCS.2013.58>
- Tan, L. (2010). Future Internet: The Internet of Things, 376–380.
- The Eclipse Foundation. (2018). Eclipse Sirius Documentation. Retrieved July 4, 2018, from <https://www.eclipse.org/sirius/doc/>
- Turcu, C. E., & Turcu, C. O. (2013). Internet of Things as Key Enabler for Sustainable Healthcare Delivery. *Procedia - Social and Behavioral Sciences*, 73, 251–256. <https://doi.org/10.1016/j.SBSPRO.2013.02.049>
- Vujović, V., Maksimović, M., & Perišić, B. (2014). Sirius : A Rapid Development of DSM Graphical Editor, 233–238.
- Wei Zhao, Chaowei Wang, & Nakahira, Y. (2011). Medical application on Internet of Things. *IET International Conference on Communication Technology and Application (ICCTA 2011)*, 660–665. <https://doi.org/10.1049/cp.2011.0751>
- Wohlin, C. (2007). Introduction to Aggregation of Case Studies Why aggregation.

- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). *Experimentation in Software Engineering*. Berlin, Heidelberg: Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-29044-2>
- Wolterink, T. (2009). Operational Semantics Applied to Model Driven Engineering. *Science*. Retrieved from <http://essay.utwente.nl/59094/>
- Wu He, Gongjun Yan, & Li Da Xu. (2014). Developing Vehicular Data Cloud Services in the IoT Environment. *IEEE Transactions on Industrial Informatics*, 10(2), 1587–1595. <https://doi.org/10.1109/TII.2014.2299233>
- Wu, Z., Itälä, T., Tang, T., Zhang, C., Ji, Y., & Liu, Y. (2012). Gateway as a Service : A Cloud Computing Framework for Web of Things, (Ict).
- Zanella, A., Member, S., Bui, N., Castellani, A., Vangelista, L., Member, S., & Zorzi, M. (2014). Internet of Things for Smart Cities, 1(1), 22–32.
- Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing : state-of-the-art and research challenges, 7–18. <https://doi.org/10.1007/s13174-010-0007-6>
- Zhou, J., Leppänen, T., Harjula, E., Yu, C., Jin, H., & Yang, L. T. (2013). CloudThings : a Common Architecture for Integrating the Internet of Things with Cloud Computing, 651–657.
-

Anexos

Anexo A: Consentimiento informado

Usted ha sido invitado a participar en el estudio titulado “Una arquitectura tecnológica CloudIoT, caso de estudio: Ambient Assisted Living”, dirigido por la Ing. Priscila Cedillo, Phd e investigadores de la Universidad de Cuenca.

El objetivo de este estudio es evaluar una arquitectura propuesta, para ello usted deberá seguir una serie de pasos y realizar una serie de modificaciones en la misma. Al final usted obtendrá un puntaje sobre 4 puntos, que permitirá entender qué tan modificable es la arquitectura. Finalmente, usted tendrá



La participación en esta actividad es voluntaria y no involucra ningún daño o peligro para su salud física o mental. Usted puede negarse a participar en cualquier momento del estudio sin que deba dar razones para ello, ni recibir ningún tipo de sanción.

Cabe destacar que no existe ningún riesgo al participar de este estudio. De participar de todo el estudio los beneficios directos que recibirá usted son los resultados de las evaluaciones y la posibilidad de ayudar a desarrollar arquitecturas más eficaces que permitan integrar componentes de hardware y de software.

Cualquier pregunta que Usted desee hacer durante el proceso de investigación podrá contactar a la **Ing. Priscila Cedillo**, Docente de la Facultad de Ingeniería de Sistemas de la Universidad de Cuenca, Teléfono: 074051000 Ext. 2359, correo electrónico: priscila.cedillo@ucuenca.edu.ec.

Yo.....con cédula de
identidad....., de
nacionalidad....., consiento en
participar en la investigación denominada: Una arquitectura
tecnológica CloudIoT, caso de estudio: *Ambient Assisted Living*
(AAL).

- Christian David Piedra García*
Pedro Alexander Tenezaca Sari



- Corresponde a la sección en donde se firma el Consentimiento.
- Incluye información clara y precisa de la investigación, relativa al propósito del estudio, modalidad de participación, riesgos y beneficios, voluntariedad, derecho a conocer los resultados, derecho a retirarse del estudio en cualquier momento, voluntariedad, derecho a conocer los resultados y confidencialidad.

Nombre y Apellido

Firma

Fecha

Anexo B: Ejercicio

El presente ejercicio tiene como objetivo medir la modificabilidad de una instancia de una arquitectura en el dominio de ambientes de vida asistidos (Ambient Assisted Living AAL) e Internet de las Cosas (IoT). Dicha instancia de deberá modificar para incluir ciertos requisitos, que permitan especificar nuevas necesidades en una solución orientada a pacientes. Por favor lea detenidamente cada una de las secciones, definiciones e instrucciones.

Sección 1:

Una arquitectura CloudIoT es una arquitectura que involucra componentes de hardware (Cosas) y software (desplegado en la nube). Esta arquitectura permite integrar Internet de las Cosas (IoT) con la nube.

Cada instancia de la arquitectura es generada de acuerdo a ciertas necesidades de un sujeto. En este caso, el dominio es Healthcare; por lo tanto, los componentes presentados en el ejercicio son de tipo médico o para asistencia médica, cabe señalar que esta solución puede ser generalizada a

cualquier dominio y pacientes que requieran soluciones de Ambient Assisted Living. En la sección 2 se definen cada uno de los componentes que pueden ser parte en la arquitectura CloudIoT.

Sección 2:

1. **Seguridad:** Sirve para definir la seguridad de un componente. Los componentes pueden ser: Fog, EntornoCloudIoT, Middleware, Adaptador o Cosa.
2. **Actor:** Son los operadores humanos del sistema. Cada componente puede tener un actor.
3. **Comunicación:** Sirve para comunicar dos componentes. Además se puede añadir un protocolo en la comunicación.
 - a. **Protocolo:** Es un sistema de reglas que permite que dos o más elementos se comuniquen. (e.j. HTTP en la capa de aplicación, WiFi en la capa de red)
4. **EntornoCloudIoT:** Sirve para definir el ambiente Cloud Computing de la arquitectura. En esta arquitectura se considera solamente Software como Servicio (SaaS).
5. **Módulo:** Un módulo es un conjunto de dispositivos que tienen una funcionalidad específica. Cada módulo puede contener elementos y comunicaciones entre ellos.
6. **Middleware:** El middleware es un habilitador importante que proporciona comunicación entre cosas heterogéneas. (e.j. se podría utilizar para comunicar dos cosas)
7. **Cosa:** Es una entidad u objeto físico que tiene un identificador único, un sistema integrado y la capacidad de transferir datos a través de una red. Además, está compuesto por sensores y actuadores.
 - a. **Sensor:** Es un tipo de dispositivo que debe estar conectado a la red de datos. Un sensor es un objeto que se puede utilizar para medir una propiedad física y convertir esa información en una señal eléctrica u óptica (p. ej. detectar el calor, el peso, el movimiento, la presión, la humedad).
 - b. **Actuador:** Es un dispositivo capaz de transformar energía en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado (p. ej. dispensar medicamentos, rotar motores).
8. **Fog:** Componente en el cual el procesamiento de los datos y las aplicaciones se concentran en los dispositivos. De esta forma, los datos pueden ser procesados localmente en un dispositivo inteligente en lugar de ser enviados a la nube. (p. ej. switches, routers, servidores embebidos, pasarela inteligente).
9. **Adaptador:** Este elemento permite que dos interfaces, en principio diferentes, puedan comunicarse. Se encargará de realizar la conversión de una interface a otra. Estas interfaces pueden hacer referencia a sistemas o servicios externos.
10. **Adaptable:** Representa la interfaz externa que se requiere comunicar (p. e.j. Sistema Metereológico)

En el Anexo 1 se puede observar cómo cada uno de los elementos de la arquitectura interactúa entre sí. La notación del meta modelo es UML (diagrama de clases). Además, existen algunos tipos de datos (p. e.j. Seguridad tiene una subcaracterística de tipo RequisitoNoFuncional. Este campo puede ser una de las opciones en la enumeración llamada RequisitoNoFuncional; es decir: confidencialidad, no repudio, integridad, autenticidad, responsabilidad o disponibilidad). En la Figura 1. se observa este ejemplo.

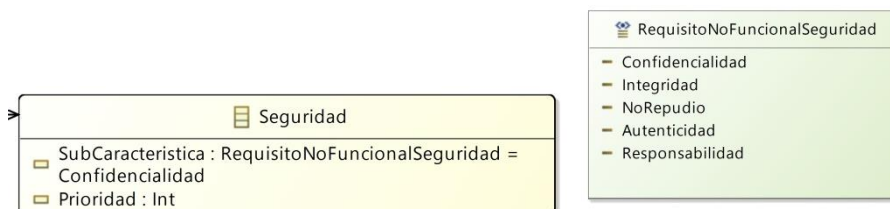


Figura 1. Seguridad y Enumeración RequisitoNoFuncional



Christian David Piedra García
Pedro Alexander Tenezaca Sari





Escenario AAL (Ambient Assisted Living)

El sistema está compuesto por sensores y dispositivos conectados al paciente, para detectar y prevenir cualquier situación de emergencia, tanto en el hospital como en el hogar. Por lo tanto, los usuarios finales de nuestro sistema son:

- **Los pacientes**, que pueden estar ubicados en el hospital o en el hogar. Las diferentes salas del hospital y también el hogar del paciente monitoreado pueden tener un conjunto de dispositivos disponibles como servicios y conectados al sistema de monitoreo de emergencia (a través de una red local o Internet, respectivamente).
- **Los profesionales de la salud o especialistas** (generalmente un médico o una enfermera), que generalmente se encuentran en sus oficinas o en las salas de emergencia. Cada especialista puede acceder a un conjunto de dispositivos disponibles como servicios que envían información relacionada con los pacientes.
- **Los centros de atención remota**, que se encargan de ayudar de forma remota a los profesionales de la salud a ayudar a los pacientes que se monitorean, con un procesamiento intermedio de la información. Los centros de atención pueden ser administrados desde el hospital o externamente, y no siempre necesitan ser parte del sistema.

El sistema está formado por los siguientes componentes:

- **Nodos de sensores** para medir los signos vitales básicos de los pacientes, como la saturación de oxígeno o la temperatura.
- **Dispositivos** con un comportamiento más complejo, utilizado para detectar (por ejemplo, aplicaciones móviles existentes para medir los dos valores de la presión arterial, alta y baja) o para controlar algunas otras acciones del paciente (p. cámara de video con un comportamiento complejo o una insignia de usuario para controlar los movimientos y algunos signos vitales al mismo tiempo). Pueden comprender varios sensores integrados e interactuando entre sí (por ejemplo, el distintivo de usuario o el lecho de un paciente, donde varios nodos de sensor están conectados, trabajando juntos).

Consideremos un caso de uso concreto el monitoreo remoto en el hogar, en el cual se monitorean periódicamente varios signos vitales de Alicia, una paciente con problemas cardíacos. Además, debido a los problemas cardíacos de Alicia, se debe implementar un mecanismo de informe de caída accidental en su casa inteligente.

El objetivo es monitorear a Alicia y, a través del análisis de la información detectada, detectar situaciones que pueden interpretarse como una caída o valores anormales en los signos vitales. Por lo tanto, si se detecta una caída, las acciones se desencadenan automáticamente, como la emisión de una alerta al centro de atención o el control de algunos de sus signos vitales (por medio de sensores siempre conectados a ella). Alguna información útil necesaria para detectar una caída es la ubicación de Alicia y la posición detectada mediante un acelerómetro. Por lo tanto, el entorno debe estar equipado con unidades de detección capaces de capturar esta información

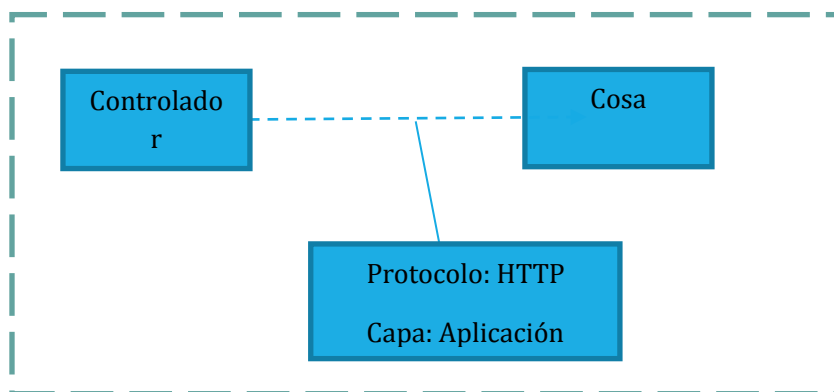
junto con actuadores capaces de contactar con el centro de atención remota siempre que se detecte una caída o un movimiento extraño.

Sección 3: Enunciado del ejercicio

Partiendo de la arquitectura propuesta en la Fig. 2, se necesitan crear nuevos componentes y posteriormente realizar algunos cambios. Para cada una de las siguientes preguntas, realizar un gráfico de cómo quedaría la arquitectura al realizar las modificaciones.

- 1) Se requiere añadir un sistema externo nuevo y comunicarlo con el módulo hospital. El sistema se llama “Clínica de Fracturas”. Además, para la comunicación entre el hospital y la “Clínica de Fracturas” se necesitan utilizar el protocolo HTTP en la capa de aplicación y Bluetooth en la capa física.
- 2) Para el hospital se ha comprado un reloj inteligente, éste va a ser utilizado por cada paciente. Este componente reemplaza al detector de caídas y al monitor cardíaco. El reloj incorpora todos los sensores y actuadores actuales en el mismo dispositivo. Para la comunicación se utiliza el protocolo HTTP. Además, se desea agregar seguridad en el reloj para proteger la **confidencialidad** de los datos.
- 3) Se desea incorporar un nuevo panel de monitoreo (dashboard) y eliminar el actual, de modo que el primero sirva para la perspectiva médica (el actor es un médico). Y el segundo para la perspectiva técnica (el actor es técnico de software). Ambos dashboards se comunican con la nube y el primero usa el protocolo MQTT y el segundo el protocolo HTTP.
- 4) Finalmente, al observar la arquitectura nos damos cuenta que se ha cometido un error y hay que corregirlo. En el punto 1) no se necesita el protocolo Bluetooth para la capa física, sino el protocolo WiFi. ¿Cómo queda finalmente la arquitectura?

Ejemplo: Se requiere agregar una nueva cosa en la arquitectura y que se comunique con el controlador mediante el protocolo HTTP en la capa de aplicación. Debido a que cada línea representa una comunicación, la cual tiene protocolos, una parte de la arquitectura quedaría de esta manera:



Como se observa en la imagen, en la comunicación se ha agregado un protocolo y se ha logrado además la comunicación del controlador con una cosa mediante una línea de comunicación.

Arquitectura Original

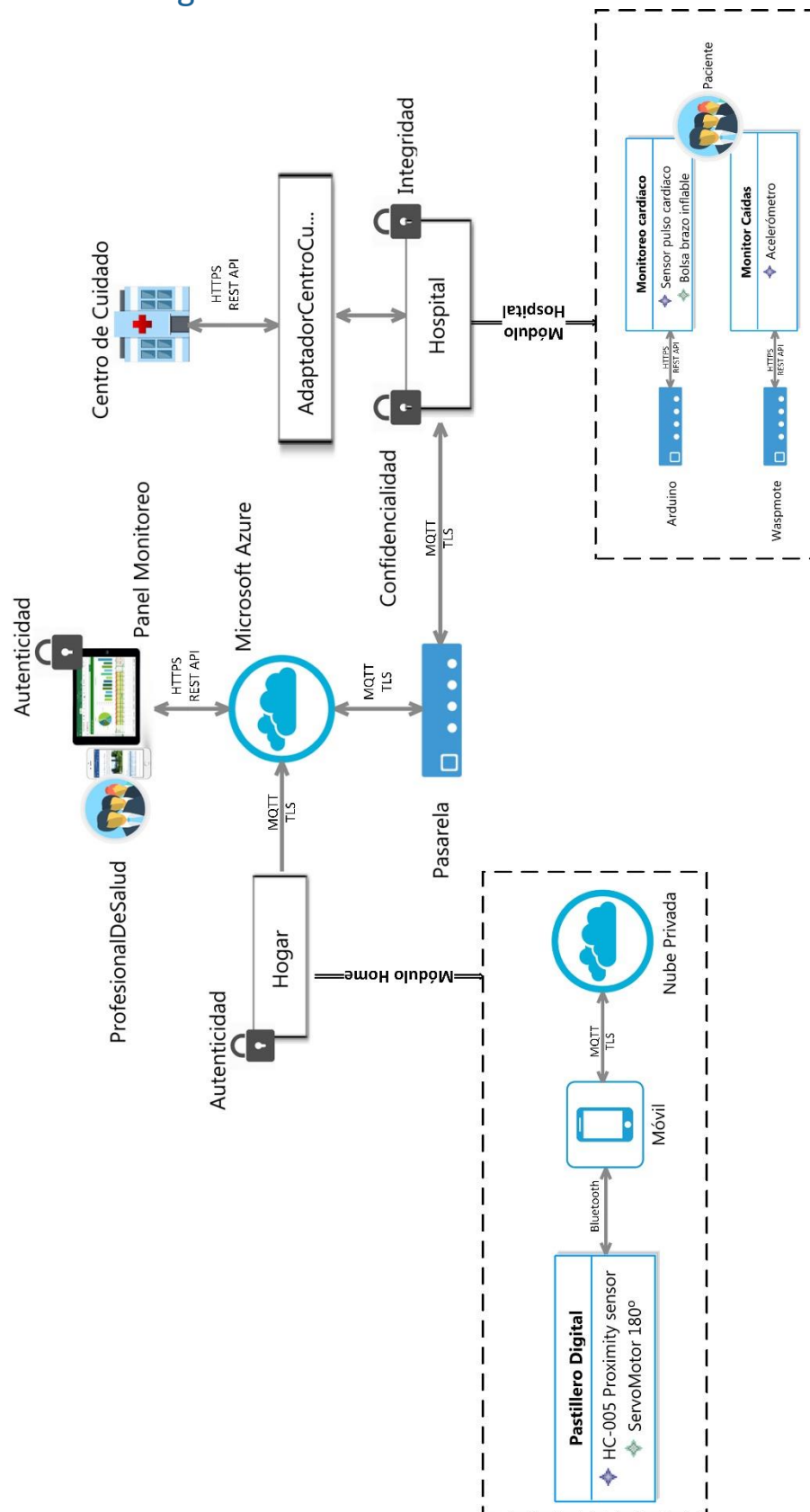


Figura 2. Arquitectura AAL propuesta



Arquitectura Modificada

Nombres:

Apellidos:

Profesión:

Hora Inicio:

Hora Fin:

Arquitectura

